EXT: Shop System

Extension Key: tt_products
Language: de
Keywords: forAdmins, forIntermediates
Copyright 2000-2010, Franz Holzinger, <franz@ttproducts.de>

This document is published under the Open Content License
available from http://www.opencontent.org/opl.shtml

The content of this document is related to TYPO3
- a GNU/GPL CMS/Framework available from www.typo3.org

Table of Contents

EXT: Shop System....1
Einleitung....3
Version....3
Abmahngefahr....3
Übersetzungen....3
Upgrade....3
Wofür ist das gut?....3
Entwickler....4
Sponsoren....4
Support....5
Benutzerhandbuch....6
Hinweis:....6
Mini-Warenkorb:....6
Berechnungsskript:....6
Rabatt:....7
Link für Produkte der letzten X Tage....7
Angebote und Highlights....7
AGB Allgemeine Geschäftsbedingungen....7
MEMO-Vormerk Seite 7
Freundschaftswerbung....7
Gutscheinpunkte System....7
Geschenk Gutscheine....8
Kauf in einer Kiste....8
Produkt Artikel mit eigenen Preisen....8
Administration....10
Installation....10
Behandlung der Kategorien....10
Behandlung der Bilder....10
Wichtige Hinweise....10
Template Marker....11
Automatisches Erzeugen von Frontend Benutzern...13
Produkt Eigenschaften:....14
Konfiguration....17
FAQ....17
Dateien....17
Beschreibung....17
Referenz....18
class.tx_ttproducts.php properties....18
Anzeigetyp (CODE)....29
CSS Konfiguration....30
Konfiguration der Artikel, Produkte, Kategorien, Seiten, Adressen und Bilder....30
Formular Eigenschaften....36
Warenkorb Konfiguration....37
Steuerungskonfiguration....38
Konfiguration für Bezahlung, Versand und Verpackung....38
Versandpreis Berechnung shippingcalc/handlingcalc....45
Pricecalc, discountprice und creditpoints Konfiguration....45
Freundschaftswerbung/Gutschein Konfiguration....47
Rechnung und Lieferschein Konfiguration....47
payment_DIBS.php....48
Bekannte Probleme....49
Prüfliste:....49
Allgemein:....49
To-Do Liste....50
Changelog....51

Einleitung

Bevor Sie mit dem Shop beginnen, sollten Sie das deutsche Tutorial oder eines der Bücher 'Der TYPO3 Webshop' und 'TYPO3 4.2 E-Commerce' durchgearbeitet haben.

Über die Extension addons_tt_products bekommen Sie neuere Beispiel Template Dateien.
Diese TYPO3 Extension läuft nur unter PHP5.

Version

Dieses Dokument gilt für tt_products Version 2.9.1. Sie erhalten die nächste Version von tt_products bereits jetzt unter http://ttproducts.de gegen eine Beteiligung an den Entwicklungskosten.

Abmahngefahr

Bitte achten Sie in den Shops und beim Verkauf über eBay sehr genau darauf, daß Sie keine Markennamen verwenden, keine nicht selbst gemachten Fotos und Textbeschreibungen verwenden und auf ein 14-tägiges Rückgaberecht hinweisen. Das Impressum auf der Homepage muss vollständig sein. Es darf keine Links auf Seiten mit Kartenausschnitten geben, und weiteres wäre zu beachten.

Viele Anwälte durchforsten in Deutschland mit ihren Helfern ständig das Internet nach fehlenden oder unzureichenden Angaben und versenden selbst bei kleinsten Verstößen massenhaft gleichlautende Abmahnungen um **einige tausend Euro** pro Fall. **Das wäre saublöd für Sie.**

```
http://www.legalershop.de/
http://www.bvdw.org/wissenspool/recht/e-commerce.html
```

Übersetzungen

Die Englische Übersetzung dieses Dokuments wird mit tt_products ausgeliefert (Pfad tt_products/doc). Eine französische Übersetzung doc_ttproducts_fr ist fertiggestellt und sucht für die Zukunft aber noch Helfer.

Upgrade

Wenn Sie ein Upgrade von einer früheren Version machen wollen, dann müssen Sie unbedingt die wichtigen Hinweise unter 'Administration' befolgen.

Wofür ist das gut?

Die Typo3 Shop Extension gibt Ihnen die Möglichkeit zu...

- Produkte mit mehreren Bildern, Details und in mehreren Sprachen aufzulisten
- Einkaufskorb
- "Bezahlen" Seite – Hier wird die Bestellung angezeigt und kann überprüft werden, bevor die Produkte endgültig bestellt werden/die Bestellung endgültig versandt wird
- für Sponsoren: Bezahlsysteme über die Payment Library – Transaction Central und Paypal
- Nachverfolgen des Zustands einer Kundenbestellung (Tracking)
- Automatische Erzeugung einer Rechnung und eines Lieferscheins
- Verschiedene Steuersätze pro Produkt, für die Zustellung und die Bezahlung
- einfache Lagerverwaltung
- Versenden einer CSV Datei für jede Bestellung an den Shop Administrator (2 auswählbare Dateiformate)
- E-Mail-Anhänge für Bestätigungs E-Mails (zum Beispiel AGB = Allgemeine Geschäftsbedingungen)
- auswählbare Produktvarianten (Farben, Größen, Staffelungen, Beschreibung, Material und Qualität)
- Zwang zum Akzeptieren der Allgemeinen Geschäftsbedingungen (AGB) durch den Kunden über eine Checkbox
- Angebote, Highlights und neu hinzugefügte Produkte
- Spezialanfertigung, Gewicht und Sperrgut (kann für die Berechnung der Versandkosten verwendet werden)
- Möglichkeit, die Zahlungsmethoden auf besondere Benutzergruppen einzuschränken
- Automatische Erzeugung eines Frontend Benutzers bei der ersten Bestellung

- Vormerkliste für Produkte – eingeloggter Benutzer kann mit der Memo-Funktion Produkt vormerken/speichern
- Rabatt in Prozent pro Benutzer
- Methoden der Preisberechnung mit Rabatten für Wiederverkäufer/Gruppen
- Anzeige der Bestellungen
- Gutscheinpunkte System: Kunden können mit jeder Bestellung Gutscheinpunkte sammeln. Die gespeicherten Gutscheinpunkte kann der Kunde bei einer neuen Bestellung einlösen . Auf Ihren Wunsch können mit diesen Gutscheinen auch nur bestimmte Produkte gekauft werden.
- Freundschaftswerbung: Wenn ein Kunde bei seiner Registrierung angibt, dass er von einem anderen (bereits existierenden) Kunden angeworben wurde, dann erhält der Anwerber eine Gutschrift in Gutscheinpunkten. Der neue Kunde erhält bei seiner ersten Bestellung einen Rabatt.
- Geschenkgutscheine: Benutzer können Geschenkgutscheine kaufen und diese per E-Mail an ihre Geschäftspartner senden. Wenn sich dieser Geschäftspartner im Frontend anmeldet, dann kann er seinen Geschenkgutschein in Gutscheinpunkte umwandeln und bereits bei der ersten Bestellung einsetzen.
- Preisstaffeln
- Feld EAN für die Eingabe des EAN Codes
- Möglichkeit zur Einschränkung der Kategorieauswahl bei der Produkteingabe im Backend
- Wertebereichprüfung des Gewichtsfeldes bis zu Miligramm
- Prüfung der Bankleitzahl, wenn die dafür erforderliche Extension static_info_tables_banks_de installiert ist
- Änderung der Anzeige bei Auswahl aus einer Varianten-Select Box
- verbesserte Kategorielisten mit mehr Markern (auch für DAM)
- beliebig viele Textbeschreibungen zu jedem Produkt (IRRE)
- Unterstützung der Payment Library 0.2.x-0.3.x
- beliebige Anzahl an Produkt-Bildern im Warenkorb
- erweiterte Gutscheine in Zusammenarbeit mit der Voucher Extension

Entwickler

– Kasper Skårhøj: Entwickler bis 1.2.7 / TYPO3 3.8

– René Fritz: Entwickler bis 1.2.7 / TYPO3 3.8

– Franz Holzinger (Projekt Leiter): Preisberechnung, Diskont Preis, Geschenkgutscheine, E-Mail Tabelle für Benachrichtigung, Fehlerkorrekturen, Rechnung, Lieferschein, mehrspaltige Listenansicht, Artikel Tabelle, mehrsprachige Tabellen

– Klaus Zierer (Projekt zk_products): mehrere Größen und Farben, Vormerkliste, AGB, CSV, automatische Registrierung der Frontend Benutzer, Seiten als Kategorien, ein beispielhaftes deutsches Template.

– Els Verberne: Gutscheinpunkte und Freundschaftswerbung System, Liste der Bestellungen

Sponsoren

– Die Entwicklung der Geschenkgutscheine wurde von Franz Holzinger durchgeführt. Es wurde von BENDOO e-work solutions aus den Niederlanden (http://www.bendoo.nl) gesponsert. Mehr Information erhalten Sie direkt bei BENDOO: hiddink@bendoo.com

– Der Verkauf von Artikeln ohne gemeinsame Attribute mit Produkten, die Kategorie Auswahl-Box, das Benennen der Bilder aus Teilen von Tabellenfeldern und der DAM Tabelle, und das Setup des Versands über die static_countries Tabelle wurde von Franz Holzinger durchgeführt. Es wurde von http://www.henrikjahn.de/ Deutschland gesponsert. Nehmen Sie mit Henrik Jahn jahn@henrikjahn.de Kontakt auf.

– pil.dk - Professionelle Internet Løsninger ApS ist ein Hauptsponsor für den Code der Payment Library Extension.

– lightimaging Bilddatenbank - wunderschöne Bilder der Natur, bildrechtefrei

– Mehrfachkategorien je Produkt mit mehrsprachiger Listenansicht, Importskripte für XML-Dateien aus Warenwirtschaftssystem, benutzerdefinierte Shop Tabellen, erweiterte Suchmaske mit Hersteller-Select-Box, Ampelsymbole über die Lieferbarkeit von Produkten. Gesponsert von Geo.net IT GmbH.

– Websolutions für Apotheke und Pharmazie

TYPO3 FullService * Konzeptionierung * Realisierung * TYPO3 Hosting *
pharmaline - Internet für die Apotheke, Apotheker Christian Platt, Kupferdreher Str 150, 45257 Essen

Support

Sie können unter http://jambage.com/index.php?id=162 und http://ttproducts.de Support, einen Installationsservice und Erweiterungen in Form von TYPO3 Extensions dazu erhalten.

Sie können alle benötigten Extensions aus dem TYPO3 TER, http://typo3.org oder von http://jambage.com/index.php?id=170 herunterladen.

Benutzerhandbuch

Hinweis:

Der Pfad zu den Dateien wurde von 'pi' auf 'pi1' umbenannt, damit tt_products jetzt gleich aufgebaut ist wie andere TYPO3 Extensionen. Daher müssen Sie nun Ihre Pfade ändern, wenn Sie diese im TypoScript Setup von älteren Versionen von tt_products verwendet haben.

Wenn Sie die Extension neu installieren oder updaten: Bitte beachten Sie die Depencies im TER (Online Repository des Extension Managers). Es muss die Table Library installiert sein.

Sie müssen das Feld 'Am Lager (St.)' in den Eigenschaften des Produktes ausfüllen. Ansonsten wird bei der Ausgabe im Frontend kein Produkt angezeigt, da der Wert dann auf 0 gesetzt ist. Null Produkte sind eben nicht vorhanden.

Vor der Eingabe der Produktpreise haben Sie die Möglichkeit zu entscheiden, ob die Preise mit oder ohne Steuern eingegeben werden (siehe TAXincluded). Alle Berechnungen und Konfigurationen werden die Preisangaben verwenden, mit denen sie in der Tabelle tt_products gespeichert sind. Sie sollten jedoch am besten die Preise inkl. Mehrwertsteuer in die Datenbank eintragen. Das erleichtert es Ihnen, die Preise gleich so eingeben zu können, wie sie auch der Kunde sehen wird. Diese Preise sollten schließlich den Konventionen der Preisgestaltung wie € 1,99 haben und nicht € 1,72 entsprechen.

Wenn Sie mit Ihrem neuen Shop beginnen, dann verwenden Sie ein Beispiel-Template aus dem Verzeichnis tt_products/template, das Template example_template_bill_de.tmpl. Sie müssen nur noch die Seiten-Id in den Links (hinter 'id=...') anpassen, damit hier die Seiten-Id des Warenkorbs ihrer Seite steht.

Der Marker ###DOMAIN### erspart Ihnen die Eingabe Ihrer URL. Das ist sinnvoll, wenn z. B. ein neuer Host verwendet wird. Im Setup des Shoptemplates muss die Domaineinstellung vorgenommen werden: plugin.domain = meinedomain.de

Wenn Sie Einträge in die Variant-Felder (Größe, Farbe ...) vornehmen, müssen Sie u. U. die selectSize und selectColors entsprechend auf 0 oder 1 setzen. Ansonsten werden in der Produktliste die Anzahlen der Produkte falsch eingetragen.

Viele Anpassungen können über den Constant-Editor vorgenommen werden. Das meiste sollte besser direkt in die Felder Setup und Constants der Templates für die Shop-Seite eingetragen werden.

tt_products basiert auf den Extensionen Table Library (table) und Static Methods for Extensions since 2007 (div2007). Wenn Sie also ein Update von tt_products durchführen, dann müssen Sie auch gleichzeitig ein Update dieser beiden Extensionen durchführen.

Mini-Warenkorb:

Neben dem "normalen" Warenkorb, der eine komplette Liste der eingegebenen Produkte in einer gewissen Sortierreihenfolge anzeigt, gibt es auch einen "Mini-Warenkorb", der nur die Anzahl der Produkte im Warenkorb und den Gesamtpreis anzeigt.

Dieser zeigt Ihnen während der Verweildauer im Shop ständig aktualisiert den Inhalt Ihres Warenkorbes oberhalb der normalen Ansicht (oder wo Sie wünschen) an. Die Stelle, an der der "Mini-Warenkorb" angezeigt werden soll, kann über Typoscript gesteuert werden. Um diesen Mini-Warenkorb verwenden zu können, fügen Sie ein neues Seiteninhaltselement "Plugin: Produkte" ein und tragen in das CODE-Feld OVERVIEW ein. Verwenden Sie die folgenden Markierungen in Ihrem tt_products Template um den Mini-Warenkorb zu aktivieren und anzupassen:

Code Listing:

```
<!-- ###BASKET_OVERVIEW_TEMPLATE### begin -->
  <div class="shop_minibasket">
     <div class="shop_minibasket_image"><!--###LINK_BASKET###-->###IMAGE_BASKET###<!--
###LINK_BASKET###--></div>
    <p>###NUMBER_GOODSTOTAL### Produkte (EUR ###PRICE_GOODSTOTAL_TAX###.-)</p>
<!-- ###BASKET_OVERVIEW_TEMPLATE### end -->


<!-- ###BASKET_OVERVIEW_EMPTY### begin -->
  <p>Keine Produkte im Warenkorb.</p>
<!-- ###BASKET_OVERVIEW_EMPTY### end -->
   </div>
```

Berechnungsskript:

Wenn Sie Ihr eigenes Berechnungsskript schreiben und verwenden wollen, dann sollten Sie immer die Preise inkl. Mehrwertsteuer eingeben. Die priceTax und priceNoTax Variablen sind Ende August 2008 gelöscht worden, weil sie aus der price Variable und den Konstanten TAXincluded und TAXpercentage berechnet werden können. Sie können die global PHP-Variablen $this->calculatedArray und $this->itemArray verwenden, um Ihre eigenen Preisberechnungen durchzuführen.

Rabatt:

Es gibt mehrere Methoden um einen Rabatt zu erhalten.

1. Verwenden des Rabatt Feldes eines Frontend Benutzers und Eingabe der Prozente für den Discount
2. Verwenden des Feldes price2 für besondere FE-User Gruppen und andere Dinge. Sie können das in TypoScript über das IF-Statement setzen.
3. Verwenden der Discountprice Berechnung. Der Rabatt hängt von der Gesamtanzahl der Produkte mit einem besonderen Preis ab. Dieser Rabatt wird für die Frontend-Benutzer der Gruppe 'discountGroupName' gewährt.

Link für Produkte der letzten X Tage

Das ist praktisch für Seiten "Was ist neu?". Erzeugen Sie einfach einen Link zur SEARCH Seite und geben Sie "newitemdays" als Parameter an. http://www.big-typo3-shop.com/index.php?id=8&newitemdays=10 Die Suche wird alle Produkte der letzten 7 Tage liefern. Sie müssen den Plugin Code LISTNEWITEMS eintragen. Sie können stattdessen auch die Konstante "newItemDays" setzen.

Angebote und Highlights

Sie müssen den Plugin Code LISTOFFERS zur Anzeige der als 'Angebot' markierten Produkte verwenden. Und mit LISTHIGHLIGHTS erhalten Sie eine Liste besonderer Produkte (Highlights).

AGB Allgemeine Geschäftsbedingungen

Sie müssen die Allgemeinen Geschäftsbedingungen akzeptieren bevor Sie eine Bestellung machen und mit dem Bestellvorgang fortfahren können. Zu diesem Zweck kann eine Seite als "PIDagb" definiert werden. Die AGBs müssen hier angezeigt werden. Sie können das auch als das Ziel eines Links über "agbtarget" setzen.

Ein Dokument (z.B. eine PDF Datei) auf dem Server, welche die AGB enthält, kann über "AGBattachment" gesetzt werden. Diese wird als Anhang an eine E-Mail mit der Bestellbestätigung an den Kunden geschickt. Die Shop Administration wird diese Datei nicht erhalten. (Sie kann stattdessen eine CSV Datei mit den Bestelldaten erhalten, falls gewünscht).

MEMO-Vormerk Seite

Ein eingeloggter Benutzer kann, ebenso wie ein nicht eingeloggter, Produkte zu einer Vormerk-Seite hinzufügen. Diese Memo Seite wird in der fe_user Tabelle oder in der FE Sitzung gespeichert. Erzeugen Sie einfach einen neue Seite mit dem Shop Plugin als Inhalt und geben Sie den Code "MEMO" ein. Sie müssen auch den Anfangspunkt setzen, wie auf den SEARCH Seiten.

Freundschaftswerbung

Ein interessierter Kunde erhält per E-Mail eine Einladung ein Shop zu besuchen und dort Waren einzukaufen. Dieser Einladung ist ein einmaliger Code beigelegt, der einer gewissen Anzahl von Gutscheinpunkten entspricht (Gutschrift). Wenn sie einen Kauf tätigen, können Sie diese Gutschrift einlösen, indem diese durch Gutscheinpunkte für den Frontend Benutzer ersetzt werden.

Wenn ein Kunde bei seiner Registrierung angibt, daß er von einem anderen Kunden angeworben worden ist, dann erhält jener Kunde Gutscheinpunkte. Der neue Kunde erhält bei seiner ersten Bestellung einen Rabatt.

Jemand (sagen wir Benutzer 1) kann einen Gutschrift Code eingeben, wenn er eine Bestellung tätigt. Dann wird Benutzer 1 5 Euro Rabatt erhalten. Dieser Gutschrift Code besteht aus dem Benutzernamen von jemand anderem (welche die E-Mail-Adresse ist), sagen wir Benutzer 2. So erhält Benutzer 2 insgesamt 5 Gutscheinpunkte. Jeder Benutzer kann nur einmal einen Gutschrift Code aus einer Freundschaftswerbung verwenden. In der fe_users Tabelle gibt es dafür ein Feld "tt_products_vouchercode", das den verwendeten Gutschrift Code (E-Mail von Benutzer 2) enthält.

Gutscheinpunkte System

Wenn ein Kunde eine gewisse Anzahl an Waren gekauft hat, dann gewinnt er dadurch Gutscheinpunkte, die auf seinem Benutzerkonto gespeichert werden. Zu einem späteren Zeitpunkt, wenn er weitere Waren kauft, kann er diese Gutscheinpunkte in Waren einlösen und beim Kauf einen besonderen Rabatt kalkulieren. Die Kunden können bei jedem Kauf Gutscheinpunkte sammeln.

```
creditpoints {
 pricefactor = 0.5
 10.type = price
```

```
 10.prod.1   = 0.02
 10.prod.101 = 0.04
 10.prod.501 = 0.06
}
```

Geschenk Gutscheine

Ein Shop Betreiber kann z.B. Gutscheine für Weinflaschen als “elektronisches Geschenk” verkaufen. Ich bestelle einen Gutschein, der z.B. 50 Gutscheinpunkten entspricht (=25 Euros), sende ihn Ihnen zu. Wenn Sie diesen Geschenkgutschein einlösen wollen, dann müssen Sie sich als Kunde des Shop registrieren und den Code auf diesem Gutschein angeben. Danach erhalten Sie die 50 Gutscheinpunkte Ihrem Konto gutgeschrieben. Mit diesen Gutscheinpunkten können Sie bestimmte Produkte einkaufen. Jemand kann z.B.: 3 Gutscheine zu 25 Gutscheinpunkten 5 Gutscheine zu 50 Gutscheinpunkten 8 Gutscheine zu 75 Gutscheinpunkten ...alle mit derselben Bestellung kaufen.

Jeder kann zum Beispiel

- 3 Gutscheine à 25 Gutscheinpunkte
- 5 Gutscheine à 50 Gutscheinpunkte
- 8 Gutscheine à 75 Gutscheinpunkte

...alles mit einer einzigen Bestellung kaufen.

Kauf in einer Kiste

In einem Shop wird Wein verkauft. Normalerweise verkaufen Sie eine Kiste von Weinflaschen (Eine Kiste kann 6 oder 12 Produkte enthalten). Der zusätzliche Service "Kauf-einer-Kiste" gibt dem Shop Kunden die Möglichkeit, sich seine eigene Kiste von 6 Flaschen einzupacken: z.B. 2 rote, 2 weiße, 2 rosé .

Unter dem Kauf-einer-Kiste Konzept wird die besondere Bearbeitung nach den Flaschen und Kisten berechnet. So wird die Bearbeitung pro Flasche mit 1,50 Euro/Flasche berechnet, die Extra Bearbeitung für 2 Kisten würde 2 x 6 x 1,50 = 18 Euro betragen.

Produkt Artikel mit eigenen Preisen

Hier wird beschrieben, wie verschiedene Größen eines Produktes zu verschiedenen Preisen verkauft werden können. Das selbe Verfahren können Sie auch mit Farben und den anderen Varianten anwenden. Achten Sie darauf, dass '
Zweite Optionspalette anzeigen' abgehakt worden ist, sonst werden einige der Felder nicht angezeigt.

Varianten:

Beschreibung (Variante 3)

Farbe (Variante 1)

2. Farbe (Variante 6)

3. Farbe (Variante 7)

Größe Länge (Variante 2)

2. Größe Breite (Variante 8)

Größe Höhe (Variante 9)

Staffelungen (Variante 4)

Material (Variante 10)

Qualität (Variante 11)

Zusätzliches (spezielle Variante 5)

1. Im TSSetup für das Template geben Sie ein

   ```
   plugin.tt_products.useArticles = 1
   ```

   Wenn sie Spalten für die Listenanzeige der Produkte verwenden, wie im Bananaguard Beispiel, dann benötigen Sie auch

   ```
   plugin.tt_products.conf.tt_products.LIST.displayColumns = 1
   ```

   Alles das teilt dem Shop mit, dass die Daten aus der Produkte Artikel Tabelle gelesen werden und daß eine Auswahlbox dargestellt wird, wenn Varianten verwendet werden.

2. Geben Sie das Vaterprodukt an.
   Das ist die erste Information, die in der Frontend Liste und in Einzelansichten angezeigt wird, z.B. in der Beschreibung und über die Bilder. Erzeugen Sie einen neuen Eintrag im Web > Liste Modul. Wählen Sie das Produkt aus..
   Sie müssen folgende Felder ausfüllen.
   - Titel
   - auf Lager (muss mindestens 1 betragen, damit es im Frontend angezeigt wird)
   - Kategorie (falls benötigt)
   - Preis (das sollte der niedrigste Preis für die billigste Größe sein – das entspricht dem billigsten Produkt Artikel, den wir als nächstes eingeben werden)
   - Größe (dasselbe gilt für Farbe/Beschreibung/Staffelung falls benötigt)
     Es ist unbedingt notwendig, um im Frontend die Möglichkeit zu haben, eine Produktvariante auswählen zu können, dass alle Größenvarianten in eine durch Strichpunkt getrennten Liste eingegeben werden, z.B.: S;M;X;XL . Diese Artikel wollen wir als nächste eingeben.
   - Bemerkung
   - Bild
   - Datenblatt (falls benötigt).

1. Anpassungen im HTML Template
   Das ist als letztes zu machen.

   - Sie müssen neue Felder in die ###ITEM_LIST_TEMPLATE### und ###ITEM_SINGLE_DISPLAY### Bereiche einfügen
   - Der folgende HTML Code muss für die Anzeige der Auswahlbox der Varianten eingefügt werden, und zwar innerhalb der Form-Tags. Außerdem wird jede Information innerhalb von ###display_variant1### oder ###display_variant2### Kommentaren nicht angezeigt, wenn diese Variante nicht verwendet wird. Das ist sehr nützlich, wenn das Wort 'Ab' vor dem ###PRICE_TAX### Wert innerhalb der ###display_variant2### Kommentare eingefügt wird. So wird z.B. 'Ab $200' wenn die billigste Größe €200 kosten soll (deshalb wird der niedrigste Preis im Vaterprodukt eingetragen. - standardmäßig wird der Preis des Vaterprodukts in der Listen- und Einzelansicht angezeigt. Der Preis der Variante wird erst im Warenkorb sichtbar, wenn eine bestimmt Größe (oder Farbe etc.) ausgewählt worden ist. Wenn keine Varianten zur Auswahl stehen, dann wird das Wort 'Ab' auch nicht angezeigt.

     Code Beispiel:

```
<!-- ###display_variant1### -->
Farbe:><select name="###FIELD_COLOR_NAME###" rows="1">###PRODUCT_COLOR###</select>
<!-- ###display_variant1### --><!-- ###display_variant2### -->
Gr&ouml;e:>< select name="###FIELD_SIZE_NAME###" rows="1">###PRODUCT_SIZE###</select>
<!-- ###display_variant2### -->
```

Administration

Im Normalfall werden die Kategorien über die Kategorietabelle und ihre Erweiterungen angezeigt. Um das zu nutzen, muß

$TYPO3_CONF_VARS['EXTCONF']['tt_products']['pageAsCategory'] = 0

eingestellt sein. Andernfalls bestimmen die Seite die Kategorie und die Kategorie ist die Unterkategorie (=1) oder die Kategorie wird nicht benutzt sondern komplett durch die Seite ersetzt (=2).

Aktivieren Sie schon jetzt die Flexforms im Backup. Dabei wird die Eingabe des Codes in das CODE Feld ersetzt durch eine graphische Oberfläche. Allerdings müssen Sie alle Codes neu eingeben.

$TYPO3_CONF_VARS['EXTCONF']['tt_products']['useFlexforms'] = 1

Installation

Installieren Sie die Extension über den Extension Manager. Wenn Sie bereits eine ältere Version von tt_products verwenden, dann ist ist in der "global" Umgebung (typo3/ext/) installiert. Wenn Sie eine neue Version ausprobieren wollen, dann sollten Sie diese unter "lokaler" Ordner (typo3conf/ext/) installieren, damit Sie die alte Version nicht dabei löschen.

Deinstallieren Sie die Extension über den Extension Manager. Danach laden Sie die eine Version Ihrer Wahl mit dem Extension Manager vom Online Repository herunter. Dann verwenden Sie im Extension Manager "Available extensions to install" und fügen das Shop System wieder hinzu.

Diese Extension läuft am besten, wenn Sie zusammen mit den Extensionen static_info_tables, sr_feuser_register, rtehtmlarea und conf_userts verwendet wird.

Behandlung der Kategorien

Es gibt mehrere Möglichkeiten ein Shop aufzubauen. Der normale Weg läuft über die Erzeugung von Sysfolder für die Produkte. Für den Aufbau eines größeren Shops mit TYPO3 wäre das etwas unpraktisch.

Wenn Sie größere Shops mit hierarchischen Kategorien anlegen wollen, dann sollten Sie die mbi_products_categories Extension und vielleicht auch nsb_cat2menu installieren. Sponsoren erhalten eine erweiterte Version von mbi_products_categories über eine mm-Tabelle, mit der man jedem Produkt eine beliebige Anzahl an Kategorien zuordnen kann. Die Anzeige von DAM Bildern wird nur damit möglich sein.

Beispiel of tt_products/ext_localconf.php:

```
$TYPO3_CONF_VARS['EXTCONF']['tt_products']['pageAsCategory'] = 0;
```

Behandlung der Bilder

Es gibt Wege die Bilder zu verwenden und anzuzeigen, die mit der DAM Extension indiziert worden sind.

Wichtige Hinweise

Das Shop ist nach Möglichkeit abwärts kompatibel zur früheren Version tt_products 1.2.7. Aber die Template Datei benötigt einige kleine Anpassungen. Am besten Sie vergleichen Ihr Template mit den Beispiel-Templates, die mit tt_products mitgeliefert werden. Im ###BASKET_TEMPLATE### müssen Sie die Zeilen

```
<input type="hidden" name="mode_update" value="1">
<input type="Submit" name="products_update" value="update basket">
```

haben.

Sie müssen im BASKET_TEMPLATE den Marker ###FIELD_NAME### in ###FIELD_NAME_BASKET### umbenennen.

Das ist kompatibel zu zk_products 1.3.2.

Einige der TypoScript Einstellungen funktionieren nicht über das Constants Feld und müssen im Setup Feld eingetragen werden. Nur die Konfigurationen, die im Konstanteneditor oder in der Datei static/old_style/constants.txt vorkommen, sind gültige Konstantenwerte. Sie müssen das 'Shop System Old Style' unter 'statische Template-Datensätze' in Ihr TYPO3 Template einbinden.

Ein negativer Wert eines Produktes bei 'auf Lager' hat keine Sonderfunktion mehr. Dies wird nun durch eine Checkbox 'immer auf Lager' ersetzt.

Ändern Sie die früheren Eingabefelder der Suche in

```
<INPUT size="30" maxlength="100" type="text" name="sword" value="###SWORD###">
```

Die Marker ###SWORDS### müssen in ###SWORD### geändert werden, ebenso der Name auf 'sword'.

Template Dateien

In den Template Dateien Ihres Shop müssen Sie die folgenden Korrekturen machen, damit Sie weiterhin verwendet werden können:

```
<input type="hidden" name="mode_update" value="1">
<input type="Submit" name="products_update" value="update basket">
```

Stellen Sie diese FORM-Attributes in Ihr BASKET_TEMPLATE.

Template Marker

Die folgenden Template Marker werden für die Template-Datei verwendet. Sie müssen ### vorher und danach eingeben. Es gibt noch einige andere Marker, die Sie bitte den Beispiel-Templates entnehmen.

Bereichsmarker:

Markierung:	Beschreibung:	Bereich:
BASKET_TEMPLATE		top
BASKET_INFO_TEMPLATE	Das wird normalerweise dazu verwendet, damit die Kunden ihre Adreßdaten außerhalb des Warenkorbs eingeben. Es hat dieselben Eigenschaften wie BASKET_TEMPLATE.	top
BASKET_ORDERCONFIRMATION_TEMPLATE	Die letzte Seite, nachdem die Bestellung verarbeitet worden ist. Wenn PIDthanks gesetzt ist, wird das nicht für die Anzeige benutzt. Aber diese Teile werden immer für die BenachrichtigungsE-Mails verwendet, wenn HTML-E-Mails eingestellt sind (orderE-Mail_htmlmail=1). siehe: BASKET_ORDERTHANKS_TEMPLATE	
BASKET_ORDERCONFIRMATION_NOSAVE_TEMPLATE	Diese HTML-Daten werden angezeigt, nachdem die Bestellung in der Datenbank gespeichert worden ist.	
BASKET_ORDERTHANKS_TEMPLATE	Anzeige einer 'Danke schön' Seite, wenn PIDthanks gesetzt ist. Für den Versand von HTML-E-Mails wird das aber nicht verwendet. siehe BASKET_ORDERCONFIRMATION_TEMPLATE	
BASKET_OVERVIEW_EMPTY	Meldung 'Ihr Warenkorb ist noch leer' für OVERVIEW	
BASKET_OVERVIEW_TEMPLATE		
BASKET_PAYMENT_TEMPLATE		
BASKET_REQUIRED_INFO_MISSING		
BASKET_TEMPLATE_EMPTY	Meldung 'Ihr Warenkorb ist noch leer'	
BASKET_TEMPLATE_NOT_LOGGED_IN	Fehlermeldung, daß sich der Benutzer nicht eingeloggt hat	
BASKET_TEMPLATE_INVALID_GIFT_UNIQUE_ID	Fehlermeldung, daß eine falsche Kennnummer für ein Geschenk-Zertifikat eingegeben worden ist	
BILL_TEMPLATE	zur Erzeugung der Rechnung	
DELIVERY_TEMPLATE	Erzeugung der Datei des Lieferscheins	
E-Mail_PLAINTEXT_TEMPLATE	Text für die E-Mail Benachrichtigung Die erste Zeile wird für das Betreff Feld verwendet.	
E-Mail_NEWUSER_TEMPLATE	E-Mail nach der Erzeugung eines Frontend Benutzers	
ITEM_LIST_TEMPLATE	Auflistung der Produkte auf der Startsetite LIST	
ITEM_LIST_GIFTS_TEMPLATE	Auflistung der Produkte Geschenk-Gutscheine	
ITEM_SEARCH		
ITEM_SEARCH_EMPTY		

Markierung:	Beschreibung:	Bereich:
ITEM_SINGLE_DISPLAY		
ITEM_SINGLE_DISPLAY_GIFT	Einzelansicht für Geschenkgutscheine	
ITEM_SINGLE_DISPLAY_NOT_IN_STOCK	Einzelansicht für Produkte, die nicht auf Lager sind	
ITEM_SINGLE_DISPLAY_RECORDINSERT	siehe displayCurrentRecord: generieren des $cObj->data	
MEMO_TEMPLATE		
MEMO_NOT_LOGGED_IN	Fehlermeldung, daß sich der Benutzer für MEMO nicht eingeloggt hat	
ORDERS_LIST_TEMPLATE	Anzeige der Liste der Bestellungen	
TRACKING_E-Mail_GIFTNOTIFY_TEMPLATE	Benachrichtigung an den Empfänger eines Geschenkgutscheins in der Verwaltung der Bestellungen	
TRACKING_ENTER_NUMBER		
TRACKING_WRONG_NUMBER		

Einzelmarker:

Beispiel für einen Wrap:

```
<!-- ###PRODUCT_LINK_DATASHEET### begin --> Datenblatt für das Produkt <!-- ###PRODUCT_LINK_DATASHEET### end -->
```

Marker:	Typ:	Beschreibung:	Bereich:
BROWSE_LINKS		Zum Durchblättern über mehrere Seiten in der Anzeige	
DELIVERYCOSTS	value	Summe der Lieferkosten und Kosten für die Bezahlung	
DELIVERY_... NOTE NOTE_DISPLAY DESIRED_DATE	value	Siehe auch PERSON_... Eingabefeld Bemerkung zur Bestellung Ausgabe Bemerkung mit für den Zeilenumbruch gewünschtes Lieferdatum	BASKET_INFO_TEMPLATE
EXTERNAL_COBJECT	value	zusätzliches Cobject	
FIELD_NAME	value in form	Name eines Feldes in einem HTML Formular	
FIELD_NAME_BASKET	value in form	Die Daten des Warenkorbs in verschlüsselter Form.	
GC1, GC2, GC3	value	globale Farben	all
GW1B, GW2B, GW1E, GW2E	wrap	globale Wraps (HTML-Umrandungen)	all
PRICE_TAX PRICE_NO_TAX PRICE_ONLY_TAX PRICE2_TAX PRICE2_NO_TAX PRICE2_ONLY_TAX	value	price/price2 des Produkts, mit oder ohne oder nur Steuer	ITEM_SINGLE
PRICE_TOTAL_TAX PRICE_TOTAL_NO_TAX PRICE_TOTAL_ONLY_TAX	value	Gesamtbetrag aller Produkte inkl. Kosten für Bezahlsystem und Versand	all
PRICE_GOODSTOTAL_TAX PRICE_GOODSTOTAL_NO_TAX PRICE2_GOODSTOTAL_TAX PRICE2_GOODSTOTAL_NO_TAX	value	total sum of all items for price or price2	all

Marker:	Typ:	Beschreibung:	Bereich:
PRODUCT_LINK_DATASHEET	wrap	Link auf die Datenblatt Datei, die in der Konstante file.datasheetIcon gesetzt worden ist. **Beispiel:** <!-- ###PRODUCT_LINK_DATASHEET1### begin --> ###GW1B######P_DATASHEET### 1: ###GW1E### <b>###PRODUCT_DATASHEET1###</ b>###ICON_DATASHEET###</p> <!-- ###PRODUCT_LINK_DATASHEET1### end --> <!-- ###PRODUCT_LINK_DATASHEET2### begin --> ###GW1B######P_DATASHEET### 2: ###GW1E### <b>###PRODUCT_DATASHEET2###</ b>###ICON_DATASHEET###</p> <!-- ###PRODUCT_LINK_DATASHEET2### end -->	
ORDER_STATUS_TIME, ORDER_STATUS, ORDER_STATUS_INFO, ORDER_STATUS_COMMENT	value	Daten zur Bestellung	TRACKING_DISPLAY_INFO
PERSON_... NAME, ADDRESS, TELEPHONE, FAX, E-Mail, COMPANY, CITY, ZIP, STATE, COUNTRY	value	Felder für die Adreßinformation ein Präfix muß vorangestellt sein PERSON_... der Kunde der Bestellung DELIVERY_... der Empfänger der Bestellung	
PID_TRACKING	value	Die Seitennummer für den Rückverfolgung über den Status der Bestellung	
STATUS_CODE_60	wrap	wird verwendet, damit das Zusenden von E-Mails an den Empfänger eines Geschenkgutscheins erst nach der Bezahlung erscheint	TRACKING_DISPLAY_INFO
STATUS_OPTIONS	value	Auswahlmenü für Zustände in der Bestellungsverwaltung	ADMIN_CONTROL inside TRACKING_DISPLAY_INFO
SHOPADMIN_E-Mail	value	E-Mail-Adresse des Shop-Adminstrators	all

globale Marker

Die globalen Marker sind für alle Subparts erlaubt. Es können neue Marker definiert werden.

Example:

```
plugin.tt_products.marks {
   mein_marker = Kaufen Sie bald wieder bei uns ein!
}

###MEIN_MARKER###
```

Property:	Data type:	Description:	Default:
marks	array	selbst definierte Marker	
markerFile *Constants:* file.markerFile	resource / *array of string*	Die Marker Datei mit den Übersetzungen. **Beispiel:** `plugin.tt_products.markerFile = EXT:tt_products/marker/de.locallang.xml`	

Automatisches Erzeugen von Frontend Benutzern

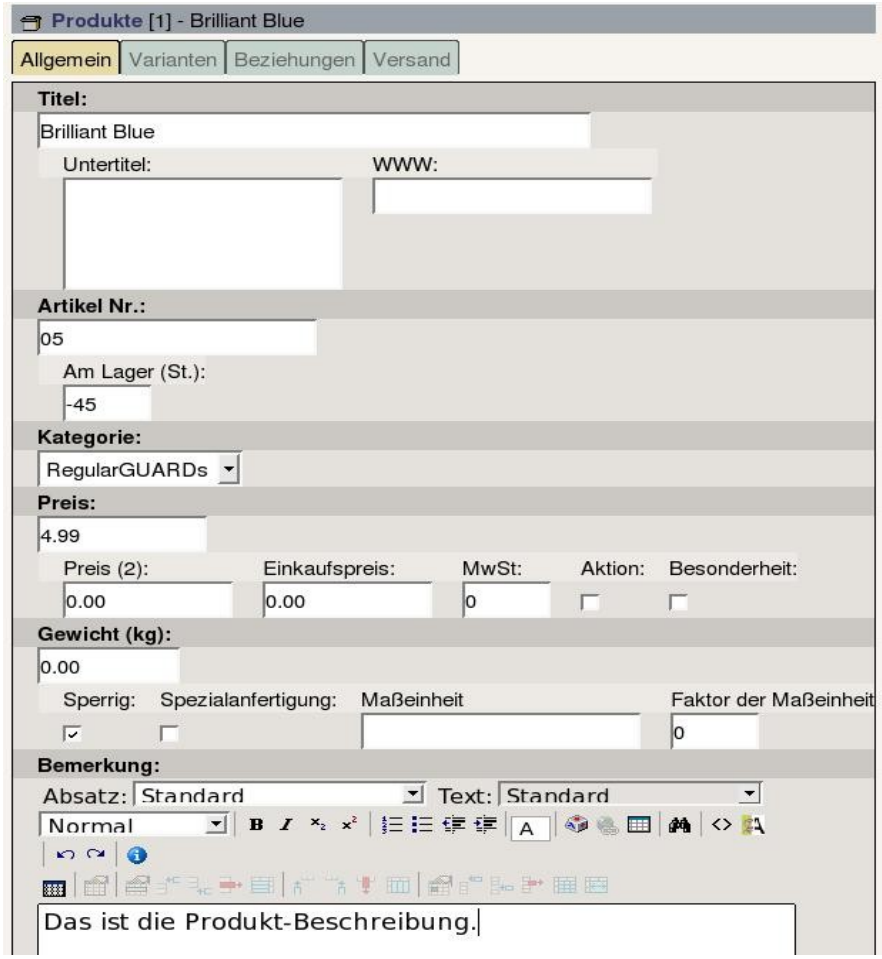
Es ist möglich nach jeder Bestellung automatisch Frontend Benutzer anzulegen. Sie müssen "createUsers" auf "1" setzen, die PID des Sysfolder als PIDuserFolder im Setup Feld eintragen. Dann müssen Sie die memberOfGroup auf die ID Ihrer Frontend Benutzergruppe setzen. Nach der ersten Bestellung erhält der Kunde eine E-Mail mit seinen Zugangsdaten. Diese E-

Mail wird den Namen des Accounts enthalten, welcher ident mit der E-Mail Adresse ist, und sein automatisch erzeugtes Paßwort.

Wenn Sie kein einzelnes Textfeld für die Adresse haben wollen, sondern eine für die Straßenbezeichnung und Hausnummer, und zusätzliche Felder für PLZ, Stadt und Land, dann müssen Sie die Template-Datei entsprechend anpassen. Wenn Sie static_info_tables installieren und "useStaticInfoCountry=1" setzen, dann wird das kleine Feld static_info_country der fe_users Tabelle anstelle des country Feldes verwendet. Das ist notwendig, wenn Sie eine Select Box zur Auswahl des Landes machen.

Produkt Eigenschaften:

Die Produkteigenschaften werden im Systemfolder für die jeweiligen Produkt im Backend eingegeben.



-

Farbe, Größe, Zusätzliches und Staffelung

Damit Sie Variationen von Produkten eingeben können, müssen Sie die Werte durch Strichpunkt getrennt eingeben. So müssen Sie für T-Shirts mit verschiedenen Farben und Größen "rot;grün;blau" als Variante 1 und "S;X;XL;XXL" als Variante 2 eingeben. Farbe (Variante 1) und Größe (Variante 2): geben Sie hier Werte mit den Titeln ein, die durch Strichpunkte 'colon ';' getrennt werden wie z.B.

```
M;L;XL;XXL
```

Ändern Sie das Template, damit das unterstützt wird:

```
<!-- ###display_variant1### -->
###GW2B### Color: ###GW2E### <select style="font-size: 10px"
name="###FIELD_COLOR_NAME###" rows="1">###PRODUCT_COLOR###</select><br>
<!-- ###display_variant1### -->
<!-- ###display_variant2### -->
###GW2B### Size: ###GW2E### < select style="font-size: 10px"
name="###FIELD_SIZE_NAME###" rows="1">###PRODUCT_SIZE###</select><br> <!-- ###display_variant2### -->
```

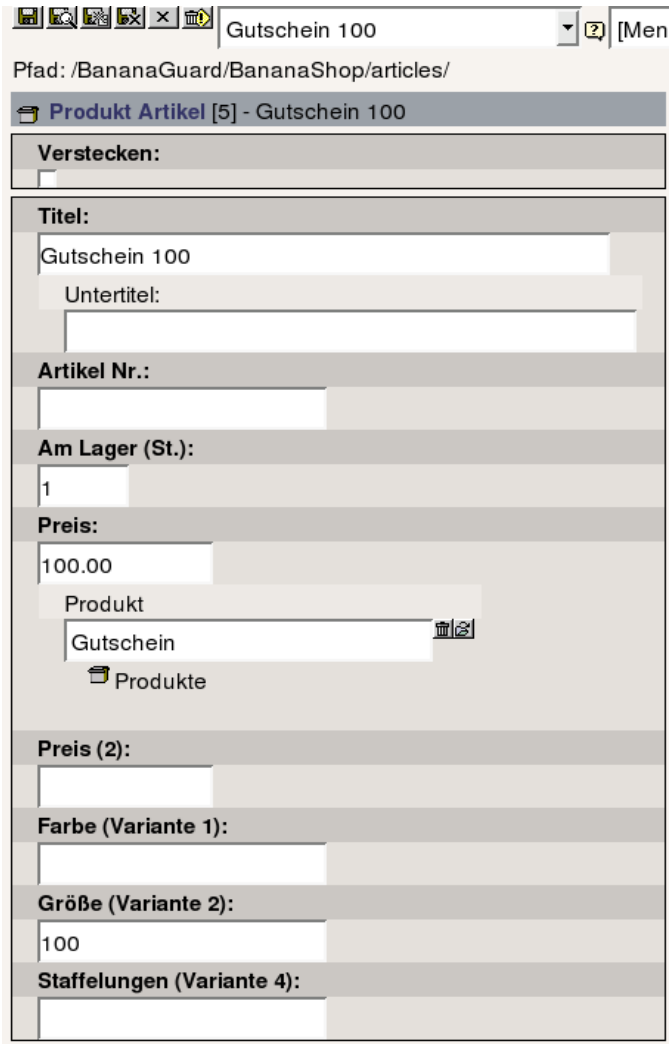
Wenn Sie die Farben und Größen nicht auswählbar machen wollen, dann müssen Sie im Template Setup selectColor oder selectSize auf 0 setzen und im Template selbst die entsprechende ###display_variant### entfernen. Sie können diese Marker jedoch zur Darstellung der Farben verwenden, auch wenn keine unterschiedlichen Farben gewählt werden können. (dann nur die Markierungen ###PRODUCT_COLOR### und ###PRODUCT_SIZE### verwenden). Denn im Setup wurden

ja Farben und Größen auf 0 gesetzt.

Der Feldname aller Felder in der Listenansicht und der Warenkorbansicht wird mit dem Marker ###FIELD_NAME### angesprochen. Dieser Marker wird intern im Shop durch einen Feldnamen ersetzt, damit das Fels ausgelesen und richtig zugeordnet werden kann.

Wenn Sie Produkte mit verschiedenen Farben, Größen und Staffelungen nach Mengen haben, dann müssen Sie im Bereich BASKET_TEMPLATE den Marker ###FIELD_NAME### durch den Marker ###FIELD_NAME_BASKET### ersetzen.

In der Listenansicht des Webmoduls können Sie nun die bereits vorhandenen Produkte um die Varianten, z.B. Farbe, ergänzen. Sie wählen dazu die Tabelle Produkt Artikel aus.



Sie müssen im Template Setup useArticles=1 setzen, um die Artikeleigenschaften zu verwenden.

Wenn Sie bereits Produkte angelegt haben, diesen bereits in Produkteigenschaften verschiedene Farben zugeordnet haben, verfügen Sie jedoch nur über eine Artikelnummer und einen Preis etc. zu diesem Produkt. Sie benötigen jedoch für Produkte z. B. unterschiedlicher Farbe auch unterschiedliche Artikelnummern und vielleicht Preise etc. Diese Zurodnungen erfolgen in den Artikeleigenschaften (Webmodul, Liste, Produkte Artikel – Tabelle). Der Vorteil ist, dass Sie die "neuen" Produkte nicht neu anlegen müssen, sondern die Eigenschaften vererbt werden. Sie geben also nur noch die Farbe, neue Artikelnummer und evtl. neuen Preis ein. Alle anderen Angaben zum Produkt werden übernommen. Lassen Sie ein Feld leer, so wird der Feldinhalt aus den Produkte Eigenschaften übernommen.

Es macht einen Unterschied, ob ein Produkt mit verschiedenen Farben oder Größen in der Produktliste oder im Warenkorb angezeigt wird. Normalerweise wird die Farbe und Größe eine Produktes über eine Auswahlbox ausgewählt. In der Produktliste wird die Anzahl für die erste Farbe/Größe angezeigt. Im Warenkorb jedoch werden die Produkte in allen Varianten einzeln aufgelistet. Die Listes der Produkte im Warenkorb und in der Auswahlseite sind also beinahe gleich, aber sie werden anders gehandhabt. Sie müssen selectSize und selectColor deaktivieren, wenn Sie keine Auswahlboxen in der

Produktliste haben wollen.

Gewicht, Sperrgut und Spezialanfertigung

Jedes Produkt kann ein Gewicht haben und als Sperrgut deklariert werden (Ja/Nein). Das Gesamtgewicht der bestellten Artikel wird berechnet und kann in der Preisberechnung für die Versandkosten berücksichtigt werden.

Wenn ein Produkt als Sperrgut markiert worden ist, dann wird bei dessen Anzeige eine Warnung als 'Sperrgut' angezeigt. ('bulkilyWarning' im Setup). Durch das Setzen von 'bulkilyAddition' im Setup können Sie zusätzliche Versandkosten für dieses Sperrgut dazuaddieren. Ein Spezialanfertigung beinhaltet keine weitere Funktionalität. Es wird nur eine Markierung (ähnlich wie bei BulkilyWarning) geschrieben. Sie können hier einen Link auf die E-Mail Seite setzen.

Beispiel:

```
specialPreparation = special preparation is possible! <A HREF="?
id=999&tt_products=###PRODUCT_ID###">Order here</A>.
```

Einfache Lagerverwaltung

Das Feld inStock kann für eine Lagerverwaltung verwendet werden. Wenn "in stock" auf "0" gesetzt ist, wird der Artikel für die Benutzer nicht mehr sichtbar sein. Nach jeder Bestellung wird die Anzahl der bestellten Artikel von "in stock" abgezogen.

Wenn ein neuer Artikel angelegt wird, wird dessen Wert standardmäßig auf '1' gesetzt. Wenn Sie 'alwaysInStock=1' gesetzt haben, dann werden diese Artikel immer verfügbar und sichtbar sein. Ansonsten wird die Anzahl der Artikel bei jeder Bestellung reduziert, auch auf negative Werte. Ein negativer Wert bedeutet, daß ein Produkt nun schleunigst beim Lieferanten bestellt werden muß.

Sie können, um einzelne Artikel immer auf Lager zu halten, die entsprechende Checkbox ankreuzen. Dies gilt aber nur für den jeweiligen Artikel. Sie können die Einheiten der Waren auf dem Lager über "inStockPieces" setzen, wie z.B. "Stück". oder Bundle. Diese Mengeneinheit kann jedoch nur einmal gesetzt werden und gilt dann für alle Produkte.

Mehrere Steuersätze

Zu jedem Produkt können Sie den Steuersatz eingeben.

Löschen des Warenkorbes

Um ein Löschfeld für ein Produkt anzugeben, muß folgendes im Template eingetragen werden:

```
<input type="submit" value="löschen" onclick="this.form.###FIELD_NAME_BASKET###.value='0'">
```

Konfiguration

FAQ

– Wenn Sie nicht über den Constants Editor arbeiten, beachten Sie die korrekte Form, die Sie in den Constants eintragen müssen:

```
plugin.tt_products {
     property = value
}
```

Dateien

Datei:	Beschreibung:
class.tx_ttproducts.php	Haupklasse zur Darstellung der Produktliste oder des Warenkorbs. Rufen Sie es von einem USER cObject über 'userFunc = user_products->main_products' auf.
products_comp_calcScript.inc	Beispiel Berechnungsskript
products_template.tmpl products_template_htmlmail.tmpl	Beispiel Templates auf Englisch. '..._htmlmail.tmpl' ist ein HTML-wrap für the versendeten HTML-E-Mails
products_template_dk.tmpl	Beispiel Template auf Dänisch
example_template_bill_de.tmpl	bananaguard.de Template mit Beispielen für Rechnung und Lieferschein auf Deutsch
products_template_fi.tmpl	Beispiel Template auf Finnisch
products_template_fr.tmpl	Beispiel Template auf Französisch
products_template_se.tmpl	Beispiel Template auf Schwedisch
product_detail.tmpl product_proefpakketten.tmpl shop-a-box.tmpl producten.tmpl	Beispiel Templates für Gutscheinpunkte-System und Freundschaftswerbung auf Holländisch
products_css_en.html	CSS basierte Template. Verwenden Sie das, wenn Sie ein barrierefreies Shop erstellen wollen.
'Bearbeitungsskripts' für den Zugriff auf externe Bezahl-Gateways	
payment_DIBS.php	Skript für eine Schnittstelle zu DIBS (Danish Internet Payment System) in Dänemark. Sie könne Sie unter http://www.architrade.com/uk/ erreichen.
payment_DIBS_template.tmpl	Template Datei für das DIBS Bezahlsystem.

[tsref:(cObject).TEST]

Beschreibung

Eingebauter Warenkorb und Produktanzeige unter TYPO3. Es beinhaltet eine freie Schnittstelle, mit der Sie Zugriff auf Ihre eigene Implementierung eines Bezahlsystems haben können.

Im Moment gibt es eine Implementierung für DIBS in Dänemark, unter www.architrade.dk.

Referenz

class.tx_ttproducts.php properties

Eigenschaft:	Datentyp:	Beschreibung:	Voreinstellung:

templateFile
Constants:
file.templateFile
	resource / *array of string*	Die Template Datei.
		Siehe Beispiel in 'tt_products/template/products_template.tmpl'
		Es kann auch ein Code Feld mit angegeben werden.
		(siehe Anzeigetyp)
		Beispiel:
		`plugin.tt_products.templateFile = EXT:tt_products/template/example_template_bill_de.tmpl`
		`plugin.tt_products.templateFile.LIST = EXT:tt_products/template/products_template_dk.tmpl`	

templateSuffix - *Setup*	string	Dieses Suffix wird an alle übergeordneten Template Subparts angehängt.	
fe	boolean	Schalter, ob FE Ausgabe vewendet wird. Die FE Ausgabe kann abgeschalten werden. Das ist für aufgerufene Skripte sinnvoll (siehe CODE SCRIPT). In diesem Fall wird keine Template Datei benötigt.	TRUE
pid_list	string /stdWrap	Die pids, aus welchen die Kategorien, Produkte und so weiter gelesen werden. Standard ist die aktuelle Seite. Hier können mehrere pids durch Beistrich getrennt eingegeben werden.	
defaultCode	string	Der Standard Code, wenn keiner für das Plugn eingetragen ist (siehe unten). Normalerweise wird das nicht gesetzt und ein Hilfe-Bildschirm wird angezeigt. **Beispiel:** `plugin.tt_products.defaultCode = HELP`	
code	string /stdWrap	siehe Kapitel Anzeigetyp	HELP
defaultArticleID	int+	Die standardmäßige Artikel Id für die Einzelansicht wird angezeigt, wenn der Link auf das Skript keinen Parameter 'tt_products[article]' gehabt hat.	
defaultProductID	int+	Die standardmäßige Produkt Id für die Einzelansicht wird angezeigt, wenn der Link auf das Skript keinen Parameter 'tt_products[product]' gehabt hat. Setzen Sie diesen Standard-Wert, wenn Sie eine Fehlermeldung wie “Falscher Parameter, GET/POST var 'tt_products[product]' wurde nicht angegeben” erhalten.	
defaultCategoryID	int+	Die standardmäßige Kategorie Id für die Listenansicht wird verwendet, wenn der Link auf das Skript keinen Parameter 'tt_products[cat]' gehabt hat. Verwenden Sie das, wenn Sie standardmäßig nur Produkte dieser Kategorie in der Listenansicht angezeigt haben wollen.	
defaultPageID	int+	Die standardmäßige Kategorie Id für die Listenansicht wird verwendet, wenn der Link auf das Skript keinen Parameter 'tt_products[pid]' gehabt hat. Verwenden Sie das, wenn Sie bei Seiten als Kategorien standardmäßig nur Produkte dieser Kategorie in der Listenansicht angezeigt haben wollen.	
defaultDAMCategoryID	int+	Siehe defaultCategoryID, aber für DAM Kategorien und den 'tt_products[damcat]' Parameter.	
productDAMCategoryID	int+	DAM category of products to be used in DAM lists.	
rootAddressID	int+	Die oberste Adressen ID, die in einer Adressenliste angezeigt wird.	
rootCategoryID	int+	Die oberste Kategorie ID, die in einer Kategorieliste angezeigt wird.	
rootDAMCategoryID	int+	Die oberste DAM Kategorie ID, die in einer DAM Kategorieliste angezeigt wird.	
rootPageID	int+	Die oberste Seiten ID von welcher diese als Kategorien gelistet werden.	
recursive	int+	Anzahl der rekursiven Ebenen für die Listenanzeige von Produkten	
domain	string	Die Url des Geschäfts. Wenn sie nicht gesetzt ist, wird sie automatisch erkannt. Damit werden die ###DOMAIN### Markierungen ersetzt.	

Eigenschaft:	Datentyp:	Beschreibung:	Voreinstellung:

altMainMarkers	*(array von strings)*	Spezifizierung von alternativen Unter-Markierungen. Für die verschiedenen Haupt Template Designs des Shop Systems.
		Diese Liste an Unter-Markierungen können Sie überschreiben:
		
		`TRACKING_WRONG_NUMBER`
		`TRACKING_ENTER_NUMBER`
		`BASKET_REQUIRED_INFO_MISSING`
		`BASKET_TEMP`
		`ITEM_SINGLE_DISPLAY_RECORDINSERT`
		`ITEM_SINGLE_DISPLAY`
		`ITEM_SEARCH`
		`ITEM_LIST_TEMPLATE`
		`ITEM_LIST_GIFTS_TEMPLATE`
		`ITEM_SEARCH_EMPTY`
		`BASKET_TEMPLATE`
		`BASKET_INFO_TEMPLATE`
		`BASKET_PAYMENT_TEMPLATE`
		`BASKET_ORDERCONFIRMATION_TEMPLATE`
		`E-Mail_PLAINTEXT_TEMPLATE`
		`BILL_TEMPLATE`
		`DELIVERY_TEMPLATE`
		
		Beispiel:
		`altMainMarkers.BASKET_TEMPLATE = BASKET_DESIGN2`
		`altMainMarkers.BASKET_TEMPLATE.wrap = ### | ###`
		
		Dieses Beispiel ändert die Haupt Markierungen der regulären Warenkorb Anzeige vom standardmäßigen ###BASKET_TEMPLATE### in das benutzerdefinierte Design ###BASKET_DESIGN2### (in derselben HTML-Template Datei)	

stdSearchFieldExt	*list of fields*	Standard Suchfelder Die standardmäßige interne Liste ist title,subtitle,note. Hier können Sie Ihre eigenen Suchfelder angeben.	
limit	int+	Max. Anzahl an angezeigten Produkten. Nicht mehr Produkte werden pro Seite angezeigt.	50
~~limitImage~~	int+	Max. Anzahl an Bildern. Die maximale Anzahl an Bildern die für ein Produkt in der Listenansicht angezeigt wird. **Veraltet.** Siehe Konfiguration der Artikel/Produkte Verwenden Sie stattdessen: `plugin.tt_products.conf.tt_products.ALL.limitImage = 1`	3
~~limitImageSingle~~	int+	Die maximale Anzahl an Bildern die für ein Produkt in der Einzelansicht angezeigt wird. **Veraltet.** Siehe Konfiguration der Artikel/Produkte Verwenden Sie stattdessen: `plugin.tt_products.conf.tt_products.SINGLE.limitImage = 1`	10

~~usePageContentImage~~	~~boolean~~	**Veraltet.** Siehe Konfiguration der Artikel/Produkte
		Verwenden Sie stattdessen:
		`plugin.tt_products.conf.tt_products.ALL.fetchImage {`
		`  type = foreigntable`
		`  table = tt_content`
		`}`	

separateImage	boolean	Normalerweise werden alle Bilder zusammen angezeigt. Mit separateImage=on kann eine Markierung ###PRODUCT_IMAGE3### für jedes einzelne Bild angegeben werden. Die Zählung beginnt bei 1.	
image	IMAGE cObject	Die Bild Konfiguration für die Einzelansicht	

listImage	IMAGE cObject	Die Bild Konfiguration für die Listenansicht
		`listImage >`
		`listImage {`
		`  altImgResource.import = uploads/media/`
		`  altImgResource.import.field = media`
		`  altImgResource.import.listNum = 0`
		`  altText.data = field:title`
		`}`
		
		Auf diese Art werden die Bilder nicht in das Verzeichnis */typo3temp/* kopiert und von dort angezeigt. (Das bereitet Schwierigkeiten mit transparenten Hintergründen), sondern direkt von /uploads/pics/.
		Die Zeile mit 'altText' bewirkt, dass der Titel als alternativer Text angezeigt wird.	

Eigenschaft:	Datentyp:	Beschreibung:	Voreinstellung:

listImageHasChilds	IMAGE cObject	nur DAM:
		Die Bild Konfiguration für die Listenansicht, falls die Seite einen Filter für eine Kategorie hat, die wieder Kinder-Kategorien hat.
		`listImageHasChilds >`
		`listImageHasChilds {`
		`  altImgResource.import = uploads/media/`
		`  altImgResource.import.field = media`
		`  altImgResource.import.listNum = 0`
		`  altText.data = field:title`
		`}`
		
		siehe listImage	

basketImage	IMAGE cObject	Die Bild Konfiguration für den Warenkorb	
datasheetIcon	IMAGE cObject	Das Symbolbild für das Datenblatt. ###ICON_DATASHEET### wird dadurch ersetzt.	
basketPic	string	URL Link auf das Warenkorb Bild	
clickIntoBasket	boolean	Falls gesetzt, wird sofort in die Warenkorbansicht gewechselt, sobald ein Produkt in den Warenkorb gegeben worden ist. Das funktioniert nur, wenn PIDbasket gesetzt worden ist.	
clickIntoList	boolean	Falls gesetzt, wird von der Einzelansicht sofort in die Listenansicht gewechselt, sobald ein Produkt in den Warenkorb gegeben worden ist. Das funktioniert nur, wenn PIDlistDisplay gesetzt worden ist.	
~~clickIntoSubmenu~~	boolean	**Veraltet.** Siehe onlyChildsOfCurrent in der Konfiguration der Kategorien ~~Falls gesetzt, werden die Untermenüeinträge der LISTCAT Kategorieliste nur für die aktuelle Kategorie angezeigt.~~	
basketMaxQuantity	int+ / string	Die maximale mögliche Anzahl eines Produkts, die in den Warenkorb gelegt werden kann. 'inStock': Nur maximal soviele Produkte können in den Warenkorb gelegt werden als sich momentan auf Lager befinden. 'creditpoint': Nur wenn der eingeloggte FE Benutzer noch ausreichend Kreditpunkte auf seinem Konto hat, darf er ein Produkt in den Warenkorb legen.	100000
quantityIsFloat	boolean	Wenn gesetzt, ist die Anzahl eine Fließkommazahl.	
noImageAvailable	resource	Die Bilddatei, die angezeigt wird, wenn ein Produkt kein Bild hat. Dieses Bild wird über das IMAGE cObject generiert, das im aktuellen Bild des Produktes angezeigt wird. Es ist eines der oberen IMAGE cObject.	
displayListCatHeader	boolean	Anzeige der Kategorie Überschrift in Listen Wenn diese Option gesetzt ist, dann werden die Kategorie Überschriften (Seitentitel) automatisch in den Produktlisten angezeigt. Das ist manchmal nicht erwünscht, wenn Sie die Überschriften selber in den Inhaltselementen von "list" ausgeben.	1
displayBasketCatHeader	boolean	Anzeige der Kategorieüberschrift im Warenkorb Wenn diese Option gesetzt ist, dann werden die Kategorieüberschriften (Seitentitel) automatisch in den Listen im Warenkorb angezeigt.	0
displayCatListType	string	Bestimmt welches HTML Tag für die Anzeige der Kategorien in der Liste verwendet wird. Verwenden Sie das neue Plugin 'Kategorien: Menü' anstelle von displayCatListType = ul.	ul
substitutePagetitle	int+	Setzen des Seitentitels in der Einzelansicht 1 ... Produkt TItel 2 ... Produkt Untertitel 12 ... Produkt TItel und Untertitel 21 ... Produkt Untertitel und TItel 3 ... Bread Crumb	
~~displayBasketColumns~~	int+	**Veraltet.** Siehe Konfiguration der Tabellen -> displayColumns	
CSS	*siehe unten*	CSS Einstellungen	
conf	*siehe unten*	Konfiguration der Tabellen	
NoSingleViewOnList	boolean	Normalerweise wird der Link auf die Einzelansicht in der Seite mit dem LIST Code generiert. Wenn Sie aber Ihre eigenen Seiten für die Einzelansicht mit dem SINGLE Code erzeugen wollen, dann müssen Sie das auf 1 setzen.	

Eigenschaft:	Datentyp:	Beschreibung:	Voreinstellung:

itemMarkerArrayFunc	function-name	Jedes Mal, wenn ein Produkt angezeigt wird, sei es im Warenkorb oder in der Einzelansicht, wird die Methode getItemMarkerArray() in user_products aufgerufen. Diese Funktion füllt ein Array aus und gibt es zurück, das sogenannte markerArray(), mit Schlüsseln/Werten für die Template Ersetzung.
		Wenn Sie hier einen gültigen Funktionsnamen eingeben (siehe Datentyp 'function-name' für Details), dann wird dieses Array anschließend als zweiter Parameter an diese Funktion übergeben. Der erste Parameter für die itemMarkerArrayFunc ist die TypoScript Eigenschaft.
		
		Parent PHP-Object Referenz:
		.parentObj property ist eine hardcodierte Referenz zum aufrufenden user_products Objekt (PHP).
		
		Beispiel:
		(vorausgesetzt, daß eine Funktione oder Klasse inkludiert ist!)
		
		`itemMarkerArrayFunc = user_addFieldsMarkerArr`
		`itemMarkerArrayFunc.simpleOption = 1`	

PIDitemDisplay	int+/*Array of integers*	PID für die Einzelansicht.
		Wenn Sie wollen, daß eine eigene Seite für die Einzelansicht von Produkten zuständig sein soll, dann geben Sie hier die PID (page-uid) an. Sie können hier Bedingungen in mehreren Zeilen angeben. Der where String kann AND Verknüpfungen enthalten. Die pid der ersten erfüllten Bedingung wird zurückgegeben.
		
		`PIDitemDisplay {`
		`10.type = sql`
		`10.where = color=red`
		`10.pid = 142`
		`20.type = sql`
		`20.where = color=white`
		`20.pid = 143`
		`}`
		
		Wenn Sie den Typ auf pid setzen, dann wird die pid des Produkt Datensatzes verwendet.
		`PIDitemDisplay {`
		`10.type = pid`
		`}`	

PIDlistDisplay	int+/*Array of integers*	PID für die Listenanzeige der Produkte Ähnlich zu PIDitemDisplay, aber hier wird die Kategorietabelle verwendet.	
PIDsearch	int+	PID für die Suchenseite Wenn Sie möchten, daß alle Produktsuchen auf eine spezielle Seite gelangen, dann tragen Sie hier die PID ein! ACHTUNG: Wenn Sie diese PID setzen, müssen alle Seiten mit Suchabfragen ein Plugin mit dem Anzeigetyp "Produkte: Suche" haben.	
PIDbasket	int+	PID für die Warenkorb Seite. Wenn Sie wollen, daß nach der Änderung der Anzahl zu einem Produkt eine Seite aufgerufen wird, dann geben Sie das hier an.	
PIDstoreRoot	int+	PID für die Wurzel des Shops. Das ist die PID der rootPage des Shops. Wenn diese nicht gesetzt ist, wird das Shop über alle Seiten ausgehend von der Root der Homepage arbeiten. Aus Performance-Gründen sollte das gesetzt sein, wenn mit Seiten als Kategorien gearbeitet wird. Verwenden Sie stattdessen besser pid_list.	
PID_sys_products_orders	int+	PID für die Datensätze in sys_products_orders. Standardmäßig erhalten sie die pid der payment (finalize) Seite.	
PIDGiftsTable	int+	PID für die tt_products_gifts Tabelle. Die Geschenk Bestellungen werden hier gespeichert.	
PIDinfo	int+	PID für die info Seite, bei welcher Name und Adresse eingegeben werden	
PIDpayment	int+	PID für die Bezahl Seite, bei welcher Name und Adresse überprüft werden und die Bestellung ein letztes Mal vom Kunden bestätigt werden muss. Von hier aus könnte auch auf einen Bezahl Gateway gewechselt werden.	
PIDfinalize	int+	PID für die Seite nach der Bestätigungsseite, wo der Benutzer seine Daten überprüft und den AGB zugestimmt hat	

Eigenschaft:	Datentyp:	Beschreibung:	Voreinstellung:
PIDthanks	int+	PID für die Dankesseite. BASKET_ORDERTHANKS_TEMPLATE wird hier verwendet. Sonst ist es ident zu PIDfinalize. Daher darf PIDfinalize nicht gleichzeitig gesetzt sein!	
PIDtracking	int+	PID für die Nachverfolgung einer Bestellung	
PIDbilling	int+	PID für die Erzeugung der Rechnung	
PIDdelivery	int+	PID für die Erzeugung des Lieferscheins	
PIDmemo	int+	Die ID der Memo Seite	
PIDagb	int+	Die PIDz einer Seite mit den Allgemeinen Geschäftsbedingungen (“AGB”) Nur wenn hier eine Seiten-Id eingetragen ist, wird geprüft, ob die AGB vom Benutzer durch das Setzen eines Häkchen vor 'Akzeptieren der AGB' akzeptiert worden sind.	0
PIDuserFolder	int+	Der sysfolder, wo neue Benutzer abgespeichert werden sollen.	116
pidsRelatedProducts	int+	Erlaubte Seiten für verwandte Produkte	
pidsAddresses	int+	Erlaubte Seiten für Adressen	
UIDstore	int+	durch Komma getrennte Liste der UIDs der Adressentabelle tx_party_addresses (oder fe_users) mit den Daten des Shops.	
paymentActivity	string	Zeitpunkt im Shop, wann die Bezahlung über ein eigenes Bezahlskript durchgeführt werden soll. Mögliche Werte: payment, finalize, verify	finalize

advanceOrderNumberWithInteger	int+ , int+	Wenn dieser Wert gesetzt ist, dann wird beim Erzeugen einer jeden Bestellung der Zähler der Bestellungen um eine Zufallszahl zwischen [erste-Integer] und [zweite Integer] erhöht, um etwas zu mogeln.
		
		Beispiel:
		`1,10 (Das wird den Zähler um einen Zufallswert zwischen 1 und 10 erhöhen)`
		`5,5 (Das wird den Zähler jedes Mal um 5 erhöhen)`	

alwaysAdvanceOrderNumber	boolean	Wenn das gesetzt ist, wird die Bestellnummer immer erhöht, und die leeren Bestellnummern werden nicht mehr wiederverwendet. Sie müssen das verwenden, wenn Sie ein Bezahlungsskript verwenden, das keine doppelten Bestellnummern akzeptiert.	
alwaysUpdateOrderAmount	boolean	Wenn das gesetzt ist, wird die bei der Eingabe angegebene Anzahl eines Produktes als neue Gesamtanzahl verwendet. Die Alternative wäre das Hinaufzählen der Anzahl um den eingegebenen Wert.	1

parseFunc	->parseFunc
	
	mit 'CSS styled content' nicht verwendet
		Wenn die Extension 'CSS styled content' installiert ist, dann müssen Sie die Einstellung in lib.parseFunc_RTE machen statt hier.
		Die Produktbeschreibungen werden mit dieser Funktion überprüft. Wenn Sie es z.B. zulassen wollen, daß im Feld 'Bemerkung' auch HTML-Tags zur Erzeugung einer Tabelle vorkommen dürfen, dann müssen Sie die parseFunc.allowTags oder auch die denyTags entsprechend setzen.
		Damit die unter RTE eingegebenen HTML-Anweisungen richtig angezeigt werden, müssen Sie Folgendes im Root Template eingeben.
		
		Beispiel:
		`keepNonMatchedTags = 1`
		`RTE.default.proc.preserveTables = 1`
		
		Beispiel:
		`parseFunc.allowTags = table,tr,td,b,i,u,a,img,br,div,center,pre,font,hr,sub, sup,p,strong,em,li,ul,ol,blockquote,strike,span,h1,h2, h3,h4,h5,h6`
		`parseFunc.denyTags = *`
			styles.content.parseFunc

categoryHeader	cObject	Erzeugt die Überschrift zu den Kategorien.
		
		Beispiel:
		
		`categoryHeader = TEXT`
		`categoryHeader.current = 1`	

Eigenschaft:	Datentyp:	Beschreibung:	Voreinstellung:

TAXpercentage	double	Prozente des Steuersatzes.
		Double Wert (!) (das bedeutet, Sie müssen den Punkt '.' statt dem Komma verwenden)
		
		Beispiel:
		`# Der deutsche Steuersatz beträgt TAX 16%:`
		`TAXpercentage = 16.00`	

TAXincluded	boolean	Setzen Sie das, wenn die Steuer in den Preisen in der Datenbank miteingerechnet ist! (... und natürlich: Löschen Sie das, wenn die Steuer nicht in den Preisen in der Datenbank enthalten ist, sondern bei der Anzeige der Preise noch dazugerechnet werden muß.) Alle Abläufe im Shop berücksichtigen diese Einstellung bei den Preisberechnungen.	1
TAXrates	string	Durch Strichpunkt/Komma getrennte List von Steuerraten. Die erste wird als Steuersatz A, nächste als Steuersatz B usw. verwendet.	
TAXmode	int+	Steuermodus 1: Die Nettosummen werden zuerst aufaddiert und erst dann wird die Steuer draufgeschlagen. 2: Die Steuer wird auf jedes Einzelprodukt draufgeschlagen. Danach werden die Bruttopreise aufaddiert.	1
priceDec	int+	Anzahl der Nachkommastellen im Preis	
priceDecPoint	string	Dezimalpunkt im Preis	
priceThousandPoint	string	Trennzeichen für die Tausender, falls gewünscht	
noZeroDecimalPoint	boolean	Wenn gesetzt, dann wird bei Ganzzahlen keine Nachkommastelle angezeigt. Statt '4.00' erscheint '4'.	
percentDec	int+	Anzahl der angezeigten Nachkommestellen bei Prozentzahlen	

priceNoReseller	int+
	S
		Preis Nummer für Wiederverkäufer. Im Moment kann hier nur 2 eingegeben werden. price2 wird aber nur dann verwendet, wenn der Wert größer als 0 ist.
		
		Beispiel:
		`[usergroup = 1]`
		`priceNoReseller = 2`
		`[global]`
		`...`	

currencySymbol	string	Währungseinheit. Es wird in der Administration verwendet.
		
		Beispiel:
		`EUR`
		`DKR`
		`USD`
		`$`	

lockLoginUserInfo	boolean	Wenn das gesetzt ist, und ein Benutzer sich eingeloggt hat, dann wird die Adreßinformation des Benutzers für die Rechnungsanschrift verwendet.	
editLockedLoginInfo	boolean	Falls das und lockLoginUserInfo gesetzt sind, dann bleiben die ausgefüllten Daten für die Bestellung editierbar. Sie müssen auch die HTML Tags entsprechend als INPUTs zur Verfügung stellen.	
loginUserInfoAddress	boolean	Wenn lockLoginUserInfo gesetzt ist, dann bewirkt dieser Schalter, dass das Adressfeld aus den Feldern address, country, zip und city aus fe_users genommen wird	

requiredInfoFields
setup
	list or array of string	Liste der Felder, die für die Adreßinformation benötigt werden. Dieses Beispiel beschreibt alle Möglichkeiten.
		
		Beispiel:
		`requiredInfoFields = name,address,telephone,fax,E-Mail,company,city,zip,state,country`
		
		Es sind weitere Unterpunkte 'billing' und 'delivery' möglich.
		
		Beispiel:
		`requiredInfoFields.delivery = name,address`	

Eigenschaft:	Datentyp:	Beschreibung:	Voreinstellung:
~~orderBy~~	~~string~~	siehe Artikel/Produkt Konfiguration **Veraltet.** Verwenden Sie stattdessen `plugin.tt_products.conf.tt_products.ALL.orderBy = title`	
~~orderByCategoryTitle~~		**Veraltet.** Verwenden Sie stattdessen `plugin.tt_products.conf.tt_products_cat.ALL.orderBy = title`	
~~orderByItemNumberSg~~	~~boolean~~	**Veraltet.** Verwenden Sie stattdessen `plugin.tt_products.conf.tt_products.SINGLE.orderBy = itemnumber`	
orderNumberPrefix	string	Präfix für die Bestellnummern. Max 10 Zeichen. Wenn dieser Text mit '%' beginnt, dann wird der Rest im PHP Datumsformat angegeben, um ein Zeit/Datum zu ergeben.	

orderEmail	*array*	Festlegung für weitere Bestätigungsemails, die über Template Suffixe, Sender und Empfänger Email Adressen beschrieben werden.
		Für 'shop' wird die orderEmail_from eingesetzt, für 'customer' die Email Adresse des Käufers.
		
		Beispiel:
		`plugin.tt_products.orderEmail {`
		`  10.suffix = factory`
		`  10.from = shop`
		`  10.to = seller1@webshop-typo3.com`
		`  10.attachment = bill`
		`  20.suffix = logistic`
		`  20.from = customer`
		`  20.to = seller2@webshop-typo3.com`
		`  20.attachment = bill`
		`}`	

orderEmail_bcc	string	BCC für die Bestätigung des Shop Betreibers	
orderEmail_from	string	Absender E-Mail Adresse für Bestätigung E-Mail an den Kunden	
orderEmail_fromName	string	Absender Name für Bestätigung E-Mail an den Kunden	
orderEmail_to	*list of E-Mail-addresses*	Eine durch Beistrich getrennte Liste der Empfänger der Bestellung E-Mail Shop und Administrator/Supervisor E-Mail Adressen müssen hier eingetragen werden!	
orderEmail_toAddress	boolean	Falls gesetzt, dann wird die Bestell Email an alle Adressen versendet, die zu einem der Produkte im Warenkorb zugewiesen sind.	

orderEmail_radio	*Feld von Radio-Button Nummer und E-Mail-Addressen*	Eine durch Beistrich getrennte Liste der Empfänger der Bestellung E-Mail, wenn eine Radio Button-Auswahl vorgenommen worden ist. Die erste Nummer ist die Radio Box und die zweite Nummer ist der ausgewählte Wert.
		
		Beispiel:
		`orderEmail_radio {`
		`  1.2 = inform@mytypo3site.de`
		`}`	

orderEmaill_toDelivery	boolean	Falls gesetzt, wird die Bestätigung E-Mail an die Zustellung E-Mail-Adresse und nicht an die Rechnung E-Mail-Adresse gesendet.	
orderEmail_subject	string	Inhalt der Betreff-Zeile, falls die erste Zeile in ###E-Mail_PLAINTEXT_TEMPLATE### leer ist.	
orderEmail_htmlmail	boolean / string	Falls gesetzt, wird die Bestellung E-Mail im HTML Format gesendet. Wenn orderEmail_htmlmail.removeImagesWithPrefix gesetzt ist, dann werden die Bilder samt ihren HTML Tags nicht in der E-Mail gesendet.	
orderEmail_apostrophe	string	Der Titel und der Absender einer Bestellung E-Mail werden damit in Hochkomma gesetzt.	
email_notify_default	boolean	Die Benachrichtigung E-Mail an den Kunden ist standardmäßig aktiviert. Er kann das selber später im Modul Bestellungsverwaltung ändern.	

Eigenschaft:	Datentyp:	Beschreibung:	Voreinstellung:

statusCodes	*array of integers*	Zustandscodes für das Modul Bestellungsverwaltung.
		Nummern über 100 löschen eine Bestellung aus der Bestellungsverwaltung.
		Nummer 0 ist der Zustand einer nicht abgeschlossenen Bestellung (und nicht abgeschlossene Bestellungen in der Datenbank können später als Müll angesehen werden...)
		Nummern zwischen 50-59 sind zur Auswahl für den Kunden.
		Nummern zwischen 60-69 sind Spezialfunktionen wie Geschenke für den Kunden.
		Nummer 1 wird dafür verwendet, wenn eine Bestellung bestägit worden ist, und deshalb von 0 auf 1 wechselt. Nummer 1 kann vom Shop Admin nicht ausgewählt werden.
		Das wird in die Markierungen ###STATUS_OPTIONS### geschrieben.
		
		Beispiel:
		`statusCodes {`
		`  1 = Bestellung vom Kunden bestätigt`
		`  2 = Bestellung ist vom Shop erhalten und akzeptiert worden`
		`  10 = Das Shop wartet auf Waren von einem Lieferanten`
		`  11 = Das Shop wartet auf die Bezahlung durch den Kunden`
		`  12 = Das Shop wartet auf Waren vom Kunden`
		`  13 = Die Bestellung wurde bezahlt`
		`  20 = Die Waren wurden an den Kunden versendet`
		`  21 = Geschenk Gutscheine wurden an den Kunden gesendet`
		`  30 = Andere Nachricht vom Shop`
		`  50 = Der Kunde bittet um die Löschung der Bestellung`
		`  51 = Meldung vom Kunden an das Shop`
		`  60 = Senden der Geschenkgutscheine an die Empfänger`
		`  100 = Bestellung versendet und geschlossen`
		`  101 = Bestellung geschlossen`
		`  200 = Bestellung gelöscht`
		`}`	

update_code	string	Das 'Paßwort', das vom Shop Administrator verwendet wird, um in die Bestellungsverwaltung des Shops im Frontend zu gelangen. Das Paßwort Formularfeld wird angezeigt, wenn ein Backend-Benutzer eingeloggt ist. Trotzdem wird hierfür nochmals ein Paßwort verlangt.	password
statusDate_stdWrap	->stdWrap	stdWrap für das Datum des Zustands **Beispiel:** `statusDate_stdWrap.strftime = %d-%m-%y %H:%M`	
orderDate_stdWrap	->stdWrap	stdWrap das Datum der Bestellung **Beispiel:** `orderDate_stdWrap.strftime = %d-%m-%y`	
usebyDate_stdWrap	->stdWrap	stdWrap des Haltbarkeitsdatums	
cardEndDate_stdWrap	->stdWrap	stdWrap für das Ende des Gültigkeitszeitraums einer Kreditkarte	
displayCurrentRecord	boolean	Wenn das gesetzt ist, werden verschiedene Einstellungen in der Anzeige der Bestellung verändert. Die Anzeige wird aus einem $cObj->data erzeugt. Wenn das gesetzt ist, wird die Untermarkierung ###ITEM_SINGLE_DISPLAY_RECORDINSERT### anstelle der gewähnlichen ###ITEM_SINGLE_DISPLAY### verwendet.	
externalProcessing	cObject	Dieses cObject kann zum Aufruf einer Funktion verwendet werden, die den Warenkorb verändert. Diese Verarbeitung kann in Abhängigkeit von den Einstellungen in einem anderen externen Shop-System erfolgen. Die Ausgabe wird in an den Anfang der Ausgabe der Warenkorbseite angehängt. Dieses cObject führt jedes mal die Methode main_products method in der Klasse user_products in class.tx_ttproducts.php aus. Sie wird immer vor jeder anderen Verarbeitung im Shop aufgerufen. Siehe Klassenbeschreibung im Code.	
externalProcessing_final	cObject	cObject für das Template der Bestellbestätigung	

Eigenschaft:	Datentyp:	Beschreibung:	Voreinstellung:
externalFinalizing	cObject	Dieses cObject kann zum Aufruf einer Funktion verwendet werden, die Einstellungen in einem externen Bestellsystem vornimmt. Dies ist eine Schwester der oben angegebenen Funktionen. Sie sollten irgendwie gemeinsam verwendet werden. Diese Funktion wird unmittelbar nach dem Aufruf der finalize-Funktion aufgerufen. Diese Funktion ist zum Beispeil dafür geeignet, um jeden externen Warenkob, der über das .externalProcessing cObject möglich ist, löschen zu können. **Achtung:** Die Ausgabe wird nirgends angezeigt.	
wrap1	-> stdWrap	Globaler Wrap 1. Das wird auf die Markierungen ###GW1B### und ###GW1E### verteilt. Geben Sie hier keinen normalen Wert ein, sondern verwenden Sie den Trennstrich zur Kennzeichnung der beiden Teile. **Beispiel:** `wrap1.wrap = <B> |</B>`	
wrap2	-> stdWrap	Globaler Wrap 2 (siehe oben) Markierungen ###GW2B### und ###GW2E###	
wrapInBaseClass	boolean *S*	Wenn gesetzt, dann wird die Ausgabe mit | umrahmt.	1

wrapInCode	boolean
	S
		Wenn gesetzt, dann wird die Ausgabe mit
		`<!-- START: tt-products-code-uid --><div id="tt-products-code-uid">`
		`|</div><!-- END: tt-products-code-uid -->`
		umrahmt. 'code' wird durch den Code des Plugins ersetzt, und uid durch das Feld 'uid' aus tt_products.
			1

selectColor	boolean	Wenn gesetzt, dann ist die Farbe eines Produkt über eine Auswahlbox wählbar.	1
selectColor2	boolean	Wenn gesetzt, dann ist die 2. Farbe eines Produkt über eine Auswahlbox wählbar.	1
selectColor3	boolean	Wenn gesetzt, dann ist die 3. Farbe eines Produkt über eine Auswahlbox wählbar.	1
selectSize	boolean	Wenn gesetzt, dann ist die Größe eines Produkt über eine Auswahlbox wählbar.	1
selectSize2	boolean	Wenn gesetzt, dann ist die 2. Größe eines Produkt über eine Auswahlbox wählbar.	1
selectSize3	boolean	Wenn gesetzt, dann ist die 3. Größe eines Produkt über eine Auswahlbox wählbar.	1
selectDescription	boolean	Wenn gesetzt, dann ist die Beschreibung eines Produkt über eine Auswahlbox wählbar.	1
selectGradings	boolean	Wenn gesetzt, dann ist die Staffelung eines Produkt über eine Auswahlbox wählbar.	1
selectMaterial	boolean	Wenn gesetzt, dann ist das Material eines Produkt über eine Auswahlbox wählbar.	1
selectQuality	boolean	Wenn gesetzt, dann ist die Qualität eines Produkt über eine Auswahlbox wählbar.	1
color1	string /stdWrap	Wert für ###GC1### Markierungen (Globale Farbe 1)	
color2	string /stdWrap	Wert für ###GC2### Markierungen (Globale Farbe 2)	
color3	string / stdWrap	Wert für ###GC3### Markierungen (Globale Farbe 3)	
nl2brNote	boolean	Wenn gesetzt, werden im Feld 'Bemerkung' (note) Zeilenumbrüche durch das HTML-Tag ersetzt.	1

Eigenschaft:	Datentyp:	Beschreibung:	Voreinstellung:
useArticles	integer	• 0: Wenn Sie keine Einträge in der Artikel Tabelle machen wollen, dann muss das auf 0 gesetzt beleiben. • 1: Wenn Sie Produkte in verschiedenen Variationen verwenden wollen und dafür eigene Preise in der Artikel Tabelle angelegt haben, dann müssen Sie das auf 1 setzen. Beim Produkt und beim Artikel müssen alle Varianten eingetragen werden. Ein Artikel wird verwendet, wenn er alle Varianten der aktuellen Auswahl über Select-Boxen abdeckt. Das Feld instock der Artikel Tabelle wird anstelle jenes der Produkt Tabelle verwendet. Die Artikel werden gemeinsam mit dem zugehörigen Produkt verkauft. • 2: Es gibt keine Varianten. Kein Produkt wird mit einem Artikel verbunden. Der Artikel wird ohne Produkt verkauft. • 3: Die Varianten eines Produkts müssen nicht mehr eingegeben werden, sondern werden von den einzeln beim Produkt zugeordneten Artikeln genommen. Einem Artikel wird kein Produkt über das Produktfeld mehr zugeordnet, sondern dem Produkt werden mehrere Artikel mit Aufpreismöglichkeit zugeordnet. Jeder Artikel kann einen Teil der Varianten für die Select-Box abdecken. Bei mehreren passenden Artikeln werden die Aufpreise addiert und die Texte aneinandergehängt.	
priceTagObj	IMAGE cObject	Bild oder cObject für die Anzeige des Preises	
usePriceTag	boolean	Wenn gesetzt, wird das priceTagObj verwendet. Wenn nicht, dann wird der Preis als normaler Text ausgegeben.	0
useStaticInfoCountry	boolean	Wenn gesetzt, dann wird das kurze fe_users static_info_country Feld als Land anstelle des country Feldes verwendet.	0
useStaticTaxes	boolean	Wenn gesetzt, dann wird die Steuer aus dem Land und der Region über die Extension static_info_tables_taxes bestimmt. UIDstore muss gesetzt sein.	0
whereGift	string	Diesee SQL Where Bedingung bestimmt, welche Produkte als Geschenke (Gutscheine) anzusehen sind.	
whereGiftService	string	Diese SQL Where Bedingung bestimmt, welche Produkte als Geschenk Services verwendet werden können.	

~~max_note_length~~	~~int+~~	siehe Artikel/Produkt Konfiguration
		Veraltet. Verwenden Sie stattdessen
		`plugin.tt_products.conf.tt_products.LIST {`
		`field {`
		`note = TEXT`
		`note.postUserFunc = tx_pmkhtmlcrop->htmlCrop`
		`note.postUserFunc.htmlCrop = 9 | ... More`
		`}`
		`}`
		
		früher:
		Die Bemerkung zu einem Produkt wir in der Listendarstellung nach max_note_length Zeichen abgebrochen. Dahinter werden noch 3 Punkte angehängt...
		Das kann aber unschön sein, wenn Sie HTML in diesem Feld verwenden. Denn wenn ein HTML-Tag nicht abgeschlossen wird, kann die ganze Darstellung chaotisch sein.
			10000

specialPreparation	string	Dieser Text wird Ihre ###PRODUCT_SPECIAL_PREP### Markierungen ersetzen.	
createUsers	Boolean	Wenn Sie das und PIDuserFolder setzen, dann wird für jeden Kunden, der nicht eingeloggt ist, und dessen E-Mail-Adresse noch nicht als Benutzer gespeichert wurde, als neuer Frontend Benutzer erzeugt. Eine E-Mail mit dem Benutzernamen (seine E-mail Adresse) und dem Paßwort wird ihm zugesendet. (siehe ###E-Mail_NEWUSER_TEMPLATE###)	0
useMd5Password	boolean	md5 Passwort Verschlüsselung aktivieren Achtung: Falls gesetzt, wird das Passwort verschlüsselt. Sie müssen dann auch Extensions wie kb_md5fepw und die newloginbox installieren.	0
outputFolder	string	Verzeichnis, wo die Rechnung und der Lieferschein angelegt werden.	fileadmin/data

Eigenschaft:	Datentyp:	Beschreibung:	Voreinstellung:
memberOfGroup	int+	Die ID der Frontend Benutzergruppe, der die neu erzeugten Frontend Benutzer angehören.	1
discountGroupName	string	Name der Frontend Gruppe für einen Discount Preis	
getDiscountPrice	boolean	Wenn gesetzt, wird der Discount Preis auch für normale Kunden gewährt, ohne daß sie zu einer Frontend Benutzergruppe in einer Discount Gruppe gehören müssen.	
AGBtarget	small	Das Target-Attribut im Link einer PIDagb	_blank
AGBattachment	resource	Wenn diese Datei existiert, wird sie an die E-Mail der Bestellbestätigung an den Kunden als Anhang mitgesendet. Der Shop Administrator wird keinen AGB als Anhang per E-Mail erhalten, aber sehr wohl eine CSV Datei mit den Bestelldaten, wenn Sie das konfiguriert haben.	fileadmin/agb.pdf
GiftAttachment	resource	Wenn diese Datei existiert, wird sie an die E-Mail zum Versenden eines Geschenkgutscheins angehängt.	
generateCSV	boolean	Erzeugt aus jeder Bestellung eine CSV-Datei. Diese Datei wird auf dem Server abgespeichert und per E-Mail an den Shop Administrator versendet (mit einer Bestellbestätigung).	0
CSVdestination	../csvorders/	Das Verzeichnis auf dem Server, wohin die CSV Dateien abgespeichert werden. Dieses Verzeichnis sollte nicht über Http zugänglich sein, weil sonst ein jeder die Daten Ihrer Kunden und der von ihnen gemachten Bestellungen einsehen könnte. Also stellen Sie dieses Verzeichnis nicht unter den Pfad von fileadmin, oder geben Sie ihm wenigstens einen schwer zu erratenden Namen.	../csvorders/
CSVfields	string	Angabe, welche Produktdaten als CSV gespeichert werden sollen. Standardmäßig werden alle Felder bis auf Bemerkung verwendet.	(Long, long list)
CSVinOneLine	boolean	Wenn Sie das auf 1 setzen, werden zusätzliche Informationen (Versand, Bezahlung, Adreßdaten, Mitteilung zur Lieferung) an das erste bestellte Produkt angehängt. Das vereinfacht den Import dieser Dateien. Wenn Sie das auf 0 gestellt lassen, wird diese Datei in Programmen wie Excel am besten angezeigt.	0
CSVnotInE-Mail	boolean	Wenn gesetzt, wird keine CSV Datei an die E-Mail angehängt.	
alwaysInStock	boolean	Wenn gesetzt, wird die Anzahl der Produkte auf Lager nach einem Kauf nicht reduziert.	1
showNotinStock	boolean	Wenn gesetzt, dann werden die Produkte auch dann angezeigt, wenn sie nicht auf Lager sind.	
notInStockMessage	string	Siehe oben. Das wird angezeigt, wenn keine Produkte auf Lager sind und wenn showNotinStock gesetzt ist.	Not in stock
warningInStockLimit	int+	Anzahl von Produkten auf Lager, ab welcher eine WarnE-Mail verschickt wird, sobald sie erreicht wird.	
inStockPieces	string	Das ist die Einheit für Produkte auf Lager inStock.	pieces
newItemDays	int+	In LISTNEWITEMS werden die in den letzten n Tagen neu hinzugekommenen Produkt angezeigt.	7
bulkilyWarning	string	Text für ###BULKILY_WARNING### für sperrige Produkte	
~~bulkilyAddition~~	~~int+~~	siehe Shipping ~~Betrag um den die Versandkosten für ein sperriges Produkt erhöht werden.~~	
~~bulkilyFeeTax~~	~~int+~~	siehe Shipping ~~Steuerbetrag in Prozent für das Versenden eines Sperrguts~~	

javaScript	*array of integers*	verschiedene JavaScripts, die für ###JAVASCRIPT_10### Markers eingefügt werden
		
		Beispiel:
		`javaScript {`
		`  10 = function addValues (a, b) { return a+b; }`
		`  20 = function multiplyValues (a, b) { return a*b; }`
		`}`	

payment / shipping	(siehe unten)	Konfiguration der Bezahlungsarten und Liefermethoden. Hier werden die Textausgaben, die Kosten und zusätzliche Berechnungen und Einstellungen für Bezahlsysteme eingestellt. Siehe Beschreibung weiter unten!	

[tsref:(script).class.tt_products.php]

Anzeigetyp (CODE)

Hier nun eine Liste der möglichen Anzeigetypen des Plugins.

Code zur Bestimmung der Darstellung und Aktion, die im Shop abläuft. Anstelle dieser Codes (Großbuchstaben) werden die Einstellungen im Shop-Plugin über Flexforms durchgeführt. Verwenden Sie die CODE-Felder nur für TypoScript Setups.

Anzeigeart:	CODE:	Beschreibung:
Produkte: Liste	LIST	Auflisten der Produkte
Produkte: Liste Geschenke	LISTGIFTS	Auflisten der Geschenke
Produkte: Liste Highlights	LISTHIGHLIGHTS	Auflisten der Produkte, die besonders attraktiv sind
Produkte: Liste Aktionen	LISTOFFERS	Auflisten der Produkte, die als Angebote gelten
Produkte: Liste Neue Artikel	LISTNEWITEMS	Auflisten der zuletzt eingetragenen Produkte
Produkte: Liste DAM	LISTDAM	Auflisten der DAM Bilder oder anderer Medien
Produkte: Einzelansicht	SINGLE	Einzelansicht eines Produkts (auch in Liste möglich) oder GET/POST 'tt_products'
Produkte: Suche	SEARCH	Suchdialog zur Suche von Produkten
Produkte: Merkliste	MEMO	Erinnerung an vorgemerkte Produkte
Produkte: zuletzt	MEMOVISIT	zuletzt besuchte Produkte
Währung: Auswahl	CURRENCY	Auswahlbox für Währungen
Warenkorb: Inhalt	BASKET	zeigt den Warenkorb mit den Produkten an, die gekauft werden sollen. Der 'BASKET' Code hat verschiedene Funktionen. Aber wenn man die spezifischen Codes INFO, PAYMENT und FINALIZE verwendet, dann kann diese Funktionalität auf mehrere Seiten aufgespalten werden
Warenkorb: Mini	OVERVIEW	Ein Mini-Warenkorb, der nur die Anzahl der Produkte enthält
Warenkorb: Eingabe Kundendaten	INFO	Eingabe der Adreßinformation
Warenkorb: Kontrolle und Bezahlung	PAYMENT	Überprüfung der Bestelldaten, Bezahlung über Bezahlsystem
Warenkorb: Bestellung abschließen	FINALIZE	schließt die Bestellung ab und versendet E-Mails Information über die Verwendung von tt_products - Seite mit Dank für die Bestellung
Bestellung: Verwaltung	TRACKING	Nachverfolgen der Bestellung, der Rechnung und des Lieferscheins
Bestellung: Rechnung	BILL	erzeugt eine Rechnungs-Datei
Bestellung: Lieferschein	DELIVERY	erzeugt eine Lieferschein-Datei
Bestellung: Liste	ORDERS	Anzeige der Bestellungen für Frontend Benutzer
Allgemein: Hilfe	HELP	Kurzanleitung
Kategorien: Liste	LISTCAT	Liste der Kategorien
Kategorien: Auswahl	SELECTCAT	Kategorien in Select Boxen
Kategorien: Menü	MENUCAT	Kategorie Menü
Kategorien: Einzelansicht	SINGLECAT	Einzelansicht für Kategorien
Kategorien: DAM Merkliste	MEMODAM	DAM Vormerkliste
Kategorien: DAM Merkliste Miniformat	MEMODAMOVERVIEW	DAM: Vormerkliste im Kleinformat
Kategorien: DAM Liste	LISTDAMCAT	Liste der DAM Kategorien
Kategorien: DAM Auswahl	SELECTDAMCAT	DAM Kategorien in Select Boxen
Kategorien: DAM Menü	MENUDAMCAT	DAM Kategorie Menü
Kategorien: DAM Einzelansicht	SINGLEDAMCAT	DAM Kategorie Einzelansicht
Artikel: Liste	LISTARTICLES	Liste der Artikel
Adressen: Liste	LISTAD	Liste der Adressen
Adressen: Auswahl	SELECTAD	Adressen in Select Boxen
Adressen: Menü	MENUAD	Adressen als Menüliste

Anzeigeart:	CODE:	Beschreibung:
Adressen: Einzelansicht	SINGLEAD	Addressen: Einzelansicht
	SCRIPT	Ausführung als PHP Skript ohne Ausgabe

CSS Konfiguration

Hier können die CSS id Namen gesetzt werden. Sie müssen eine CSS Datei im Template Setup angegeben haben, die diese ids auch verwendet.

Sie müssen auch den Namen der Tabelle im Setup mit angeben.

Der vor/zweitletzte Wert enthält den Codenamen der Ansicht. Wenn hier 'ALL' steht, dann gilt es für alle Ansichten.

Ansichten, die dem Code-Feld entsprechen:

SINGLE, LIST, BASKET

```
plugin.tt_products.CSS.ALL.file = EXT:ttproducts/template/tt_products_example.css
```

allgemein

Eigenschaft:	Datentyp:	Beschreibung:	Voreinstellung:
file	resource	Die CSS-Datei.	

```
plugin.tt_products.CSS.tt_products.LIST.row.even = 35
```

Tabelle

Eigenschaft:	Datentyp:	Beschreibung:	Voreinstellung:
row		even: Cascading Stylesheets (CSS) gerade Zeilen in der Produktliste. uneven: ungerade	
list		default: CSS für Standardeinträge in der Listenansicht current: CSS für die ausgewählten Einträge in der Listenansicht.	
~~menu~~	~~string~~	~~CSS für das Menü~~	
itemSingleWrap	wrap	HTML Teil, der die Marker ###ITEM_SINGLE_PRE_HTML### und ###ITEM_SINGLE_POST_HTML### auf Produktbasis ersetzt.	| oder <td>|</td>
itemRowWrap	wrap	HTML Teil, der die Marker ###ITEM_SINGLE_PRE_HTML### und ###ITEM_SINGLE_POST_HTML### auf Produktreihebasis ersetzt.	leer oder <tr>|</tr>

Konfiguration der Artikel, Produkte, Kategorien, Seiten, Adressen und Bilder

Der vor-/zweitletzte Wert enthält den Codenamen der Ansicht. Wenn hier 'ALL' steht, dann gilt es für alle Ansichten.

Ansichten, die dem Code-Feld entsprechen:

SINGLE, LIST, BASKET

zusätzlich mögliche Werte:

E-Mail, PAYMENT, LISTRELATED (LIST für verwandte Produkte)

Beispiel:

```
plugin.tt_products.conf.tt_products_articles.LIST.generatePath.base = fileadmin/images
plugin.tt_products.conf.tt_products.LIST.orderBy = sorting
```

Eigenschaft:	Datentyp	Beschreibung:	Voreinstellung:

generatePath	string	Pfad zu den Verzeichnissen der Bilder für generateImage
		
		Paare von Feldnamen und die Anzahl der ersten Buchstaben, die zur Bildung des Namens der Bilddatei verwendet werden.
		type ... tablefields (Tabellenfelder)
		field.*fieldname* ... Name des Tabellenfeldes
		
		Beispiel:
		`ALL.generatePath {`
		`  type = tablefields`
		`  base = fileadmin/images`
		`  field.itemnumber = 2`
		`}`
			fileadmin/img

generateImage	*array of string*	Paare von Feldnamen und die Anzahl der ersten Buchstaben zur Bestimmung des Namens der Bilddatei.
		
		type ... tablefields (Tabellenfelder), foreigntable (andere Tabelle, falls benötigt)
		field.*fieldname* ... Name des Feldes der Tabelle
		table ... andere Tabelle und deren Konfiguration um das Bild zu ermitteln
		uid_local ... verwende dieses lokale Feld der Tabelle
		uid_foreign ... verwende dieses Feld der externen Tabelle um eine Übereinstimmung zu finden
		parent ... verwende auch Feldnamen der Vater-Tabelle
		parent.depth ... Anzahl der Vaterebenen
		
		Beispiel:
		`ALL.generateImage {`
		`  type = tablefields`
		`  field.itemnumber = 6`
		`}`
		
		`ALL.generateImage {`
		`  type = foreigntable`
		`  table = tt_products_articles`
		`  uid_local = uid`
		`  uid_foreign = pid`
		`  field.itemnumber = 6`
		`}`	

imageMarker	*array of string*	Beschreibt woraus sich der Marker für das Bild zusammensetzt. Als Beispiel wird das Bild der Datei 30_P1_001.jpg verwendet. Der zweite Teil P1 und der dritte Teil 001 bilden den Marker. Als Ergebnis erhält man den Marker ###CATEGORY_IMAGE_P1_001###. Der erste Teil '30' muß in diesem Beispiel pro Datensatz verschieden sein.
		
		Beispiel:
		`ALL.imageMarker {`
		`  type = imagename`
		`  parts = 2,3`
		`}`	

limit	int+	Max. Anzahl an angezeigten Produkten. Bis zu so viele Produkte werden in einer Listenansicht angezeigt.	50

Eigenschaft:	Datentyp	Beschreibung:	Voreinstellung:
limitImage	int+	Max. Anzahl der angezeigten Bilder für ein Produkt. **Beispiel:** `plugin.tt_products.conf.tt_products.ALL.limitImage = 10`	1
orderBy	string	Liste der Felder, nach denen die Datensätze sortiert werden.	sorting

fetchImage	string	Die Bilder einer anderen Tabelle werden statt den Bildern dieser Tabelle verwendet.
		
		Beispiel:
		`plugin.tt_products.conf.tt_products.ALL.fetchImage {`
		`  type = foreigntable`
		`  table = tt_content`
		`}`	

language	*Feld von Strings*	Der Name der Sprachdatei mit Übersetzungen aus der Default Sprache in eine andere Sprache.
		type:
		• csv ... Die Werte werden mit ';' und Zeilenumbrüchen getrennt
		• noTranslation ... die Tabelle der alternativen Sprache soll nicht verwendet werden
		• field ... Die Übersetzung befindet sich in Feldern
		• table ... Die Sprachtabelle
		
		file: Pfad und Name der Datei
		field: Name des Feldes auf der linken Seite und Wert auf der rechten Seite
		marker: Wie CSV, aber die Marker innerhalb der Datenbanktabelle werden ersetzt.
		
		Beispiel:
		`[globalVar = GP:L = 1]`
		`language {`
		`  type = csv`
		`  file = fileadmin/data/EnglishCategories.csv`
		`}`
		`[GLOBAL]`
		
		Beispiel:
		`language {`
		`  type = field`
		`  field.title = subtitle`
		`}`	

image	IMAGE cObject	Das Image Objekt kann über TypoScript in andere kopiert werden und somit für mehrere Code Felder bzw. Ansichtsarten verwendet werden. **Beispiel:** `###PRODUCT_IMAGE1:M###` `plugin.tt_products.conf.tt_products.ALL.image.m {` `  wrap = | ` `  file.maxW = 320` `  file.maxH = 280` `}`	

Eigenschaft:	Datentyp	Beschreibung:	Voreinstellung:

filter	*Feld von Strings*	Falls das eingetragen ist, werden nur die Datensätze der Tabelle verwendet, die durch diesen Filter auf Basis von Tabellenfeldern oder Parametern passen.
		type:
		• param ... GET/PUT Parameter (Wert: gp) oder vordefiniert
		• regexp ... Verwenden eines regulären Ausdrucks
		• where ... Verwenden von WHERE Bedingungen
		field: Name des Feldes auf der linken Seite und Wert auf der rechten Seite
		Beispiel:
		`filter { regexp { field { title = [:alpha:]+[:blank:]+1[:blank:]+ } } }`
		`filter { where { field { bargain = 1 } } }`
		`filter { param { cat = gp } }`
		Wird alle Datensätze ausfiltern, die im Titelfeld nur Buchstaben haben und am Ende eine Ziffer 1.	

listProductsNoCat	boolean	Normalerweise werden keine Produkte unterhalb einer Kategorieliste angezeigt, solange noch keine Kategorie ausgewählt worden ist. Wenn dieser Wert auf '1' gesetzt wird, dann werden immer alle Produkte in der Listenansicht angezeigt, falls kein Kategorie-Parameter vorhanden ist und sich eine Kategorieliste auf derselben Seite befindet.	
urlparams	string	Durch Beistriche/Komma getrennte Liste von tt_products URL Parametern, die einen Wert haben müssen. Andernfalls werden keine Datensätze angezeigt. Normalerweise sollen keine Produkte unterhalb einer Kategorieliste angezeigt, solange noch keine Kategorie ausgewählt worden ist. Wenn hier nichts eingetragen ist, dann werden immer alle Produkte in der Listenansicht angezeigt, falls kein Kategorie-Parameter vorhanden ist.	

Eigenschaft:	Datentyp	Beschreibung:	Voreinstellung:
displayColumns	*Feld von Strings*	Anzahl der Spalten bei der Anzeige Anzahl an Spalten in den Ansichten. Es wird eine HTML Tabelle aufgebaut. Dazu müssen spezielle Template Markierungen verwendet werden. Die ITEM_SINGLE_PRE_HTML und ITEM_SINGLE_POST_HTML müssen vor und nach den <TD>-</TD> Tags stehen, damit die Tabelle richtig erzeugt wird. Als Zahl wird zuerst die Hierarchieebene der Kategorie genannt. **Beispiel:** `displayColumns {` `  1 = 3` `}` **Beispiel:** `<!-- ###ITEM_SINGLE### begin-->` ` ###ITEM_SINGLE_PRE_HTML###` ` <td height="150" bgcolor="###GC1###" valign="bottom" align="center">` ` <!--###LINK_ITEM###-->` `###GW1B###<b>###PRODUCT_TITLE###</b>###GW1E###` ` <!--###LINK_ITEM###-->` `  ` ` ###PRODUCT_IMAGE###  ` ` ###GW1B### item count: ###GW1E### <input size="3" maxlength="4" type="text" name="###FIELD_NAME###" value="###FIELD_QTY###">` ` <p> </td>` ` ###ITEM_SINGLE_POST_HTML###` ` <!-- ###ITEM_SINGLE### end -->`	

displayHeader	string	Jede Liste von Elementen kann einen Kopfteil (Header) für die Kategorie haben. Normalerweise wird zuerst der Kopfteil und danach alle Elemente dieser Kategorie angezeigt. Aber manchmal soll der Kopfteil immer mit jedem Element zusammen wiederholt werden.
		Werte:
		● always ... Der Block bestehend aus Kategorie und Element wird mit jedem Element wiederholt. Es werden nicht wie sonst üblich, die Elemente der gleichen Kategorie zusammengefasst.
		● current ... Es wird nur die aktuelle Kategorie verwendet und nicht eine 2. Kategorie eines Elements. Damit gehören alle angezeigten Elemente zu dieser einen Kategorie, die gerade aktiv ist.
		
		Example:
		`displayHeader {`
		`  1 = always`
		`}`	

requiredFields	string	Felder, die aus der Datenbanktabelle gelesen werden müssen, auch wenn es keinen Marker im aktuellen Template Subpart dafür gibt.	

special	*Feld von Strings*	Spezielle Behandlung für die Tabelle. Nur bei Kategorie Parametern verwendet.
		
		• all ... Diese uid bedeutet, daß alle uids im Parameter aktiv sein sollen. 'all=all' bedeutet, dass immer alle uids aktiviert werden. Der Parameter wird also ignoriert.
		• no ... Diese uid bedeutet, daß diese uid im Parameter nie aktiv sein soll.
		
		Example:
		`special {`
		`  all = 1`
		`}`
		
		Tabelle tt_products_cat:
		Das bewirkt, daß die Produkte aller Kategorien angezeigt werden, wenn der Parameter 'tt_products[cat] = 1' gesetzt worden ist. So werden anstelle der Produkte von Kategorie 1 nun alle Produkte aller Kategorien aufgelistet.	

onlyChildsOfCurrent	boolean	Wenn gesetzt, werden die Kinder Elemente (z.B. Unterkategorien) nur für ein aktuelles Element gezeigt (z.B. die aktuelle Kategorie).	

Eigenschaft:	Datentyp	Beschreibung:	Voreinstellung:
field	*Feld von cObject*	Verwendung von cOjects um einige Felder darzustellen. Normalerweise wird jedes Feld so angezeigt, wie es in der Datenbank Tabelle gespeichert worden ist. Nur die image Felder werden als IMAGE Bilder dargestellt. **Beispiel:** `field { title = COA title { 10 = TEXT 10.value = ###PRODUCT_TITLE### in sizes ###PRODUCT_SIZE### 10.wrap = Title: - | - } }`	
fieldIndex	int+/*Array of integers*	Index für Felder mit mehrfachen Inhalten. Das wird vorerst nur verwendet, um anzugeben, welche Artikelbilder welche Bilder des Produktes überschreiben sollen. Links steht der Index des Bildes (Beginn mit 1) und rechts die Zielnummer des Bildes. Der Marker für das 1. Bild wäre dann ###<TABELLENKÜRZEL>_IMAGE2###. `fieldIndex { image { 1 = 2 } }`	
controlFields	*Feld von cObject*	Steuerfelder newitemdays ... Name des Feldes, welches mit newItemDays verwendet wird. **Beispiel:** `controlFields { newItemDays = starttime }`	newItemDays -> starttime
displayFields	*array of cObject*	Anzeigefelder Einstellungen für die Anzeige einiger Felder **Beispiel:** `displayFields { note = RTEcssText }`	
fieldLink	*array of typolink*	Die Definition der Links auf Feldinhalte. **Example:** `conf.tt_products.ALL { fieldLink { datasheet.ATagParams=class="datasheetClass" } }`	
fe_users	*array of string*	FE Benutzer Felder für Datumsgrenzen wie Altersbeschränkungen date_of_birth.period ... Diese Zeitdauer muss seit der Geburt des FE Benutzers vergangen sein, damit die Produkte mit uid 12 oder 24 im Warenkorb erlaubt sind. **Example:** `fe_users { date_of_birth.period.y = 12 where = uid IN (12, 24) }`	

Eigenschaft:	Datentyp	Beschreibung:	Voreinstellung:

view	*array of string*	Konfiguration der Anzeige von CODE Inhalten
		
		sortSelect ... Feld von Form Elementen
		filterSelect ... Filter Funktionen
		
		Beispiel:
		`view.sortSelect {`
		`1 {`
		`label = sort by`
		`valueArray {`
		`10.label = title`
		`10.value = 1`
		`10.field = title`
		`}`
		`}`
		`}`
		
		browser ... Konfiguration des Datensatz Browsers mit Typ div2007
		showFirstLast ... wenn der Link für das erste und letzte Element angezeigt werden soll
		
		Beispiel:
		`view.browser = div2007`
		`view.browser {`
		`browseLinksWrap = <div class="browseLinksWrap">|</div>`
		`disabledLinkWrap = <span class="disabledLinkWrap">|</span>`
		`inactiveLinkWrap = <span class="inactiveLinkWrap">|</span>`
		`activeLinkWrap = <span class="activeLinkWrap">|</span>`
		`disabledNextLinkWrap = <span class="pagination-next">|</span>`
		`inactiveNextLinkWrap = <span class="pagination-next">|</span>`
		`disabledPreviousLinkWrap = <span class="pagination-previous">|</span>`
		`inactivePreviousLinkWrapn = <span class="pagination-previous">|</span>`
		`}`	

subpart	*array of string*	Damit bestimmt man das Verhalten bei der Anzeige der Subpart Marker.
		
		show ... default Wenn der Datensatz einer default Kategorie immer angezeigt werden muss, auch dann wenn keine Produkte zu dieser Kategorie gefunden werden. Das wird dann verwendet, wenn man für jede Seite eine Default Kategorie festgelegt hat und erreichen will, dass immer ein Kategorie Titel und Bild angezeigt wird.
		
		Example:
		`subpart.ITEM_CATEGORY {`
		`show = default`
		`}`	

Formular Eigenschaften

Nur für Setup. Es gibt mehrere Formulare, die konfiguriert werden können. Geben Sie den Code nach dem 'form' an.

Example:
`plugin.tt_products.form.SEARCH.data.name = ShopSearchForm`

Eigenschaft:	Datentyp	Beschreibung:	Voreinstellung:
data	string/->stdWrap	Daten, aus denen das Formular zusammengesetzt wird. name ... Name des Formulars. Falls leer, wird der gesamte Subpart (Unterbereich) nicht gezeichnet.	*abhängig vom Codefeld*

Eigenschaft:	Datentyp	Beschreibung:	Voreinstellung:

dataArray	*Feld von Form Elementen*	Jeder Eintrag im dataArray ist numerisch und hat die 4 Haupteigenschaften '*label*', '*type*', '*value*' und '*required*'. 'label' und 'value' haben stdWrap Eigenschaften.
		params ... Zusätzliche Parameter angeben
		
		Beispiel:
		`form.SELECTAD.dataArray {`
		`10.label = Hersteller:`
		`10.type = tt_products[address]=select`
		`10.params = onchange = "submit();"`
		`}`
		
		Enhancement for the selection of variants with selectColor, ... specification.
		
		`form.ALL.selectColor.dataArray {`
		`10.params = onchange = "submit();"`
		`}`	

image	IMAGE	Bild für die Ausgabe	

imageImport	*array of fields*	Bestimmung der Bilder anhand der relativen Position einer Variante
		
		Beispiel:
		`form.ALL.selectColor.imageImport {`
		`10.sql.where =`
		`10.prod.0 = earth-round.gif`
		`10.prod.1 = marigold-round.gif`
		`10.prod.2 = delft-round.gif`
		`}`	

layout	string	Hierüber wird bestimmt, wie die Eingabefelder und andere Marker relativ zueinander eingefügt werden.. **Beispiel:** Das ersetzt "###INPUT###" mit dem Input HTML Tag und "###IMAGE###' mit der Bildausgabe. `<tr><td>###INPUT###</td><td> ###IMAGE###</td></tr>`	

Warenkorb Konfiguration

Hier kann das Verhalten des Warenkorbs eingestellt werden.

Beispiel:

```
plugin.tt_products.basket.minPrice {
    type = price
    collect = goodstotal
    value = 250
}
```

Eigenschaft:	Datentyp	Beschreibung:	Voreinstellung

minPrice	*Feld von Strings*	Mindestbestellwert, den die Produkte erzielen müssen, damit die Bestellung angenommen wird. Es sollen zum Beispiel nur Bestellungen ab einem Gesamtwert von 250 Euro angenommen werden. Wenn in einem Produkt 'kein Mindestbestellwert' angekreuzt ist, dann wird es nicht mitgezählt.
		
		Beispiel:
		`minPrice {`
		`type = price`
		`collect = goodstotal`
		`value = 250`
		`}`	

Eigenschaft:	Datentyp	Beschreibung:	Voreinstellung

view	*array of string*	Konfiguration der Warenkorbanzeige
		
		showAmount … basket ... wenn die Anzahl der Produkte im Warenkorb auf der Einzel- und Listenansicht angezeigt werden soll.
		… 0 … wenn die Anzahl bei den Produken immer 0 ist (für Caching)
		
		input … Eingabefeld 'in den Warenkorb' als dataArray für den Marker ###BASKET_INPUT###
		Radio Buttons für die Listenansicht der Produkte
		Eingabe der Namen der Radio Buttons
		where: SQL WHERE Bedinung für die Produkte
		checked: voreingestelltes Produkt
		
		Beispiel:
		`basket.view.input {`
		`  10.label = einkaufen:`
		`  10.type = radio`
		`  10.where = uid IN (12,13,14,15)`
		`  10.checked = 12`
		`  10.name = Radiobox1`
		`  10.params =`
		`}`
			showAmount = basket

activity	*array of string*	Einstellung für Aktivitäten.
		Es gibt diese möglichen Aktivitäten:
		basket, info, payment, verify, finalize, overview
		
		check: (Prüfung)
		address … ob die verlangten Felder ausgefüllt worden sind (siehe requiredInfoFields)
		agb … ob die AGB Checkbox angeklickt worden ist
		basket … ob sich etwas im Warenkorb befindet
		
		Beispiel:
		`basket.activity.payment {`
		`  check = address,agb,basket`
		`}`	

store	*boolean*	Wenn auf 0 gesetzt, dann wird kein Warenkorb gespeichert. Das ist nur dann notwendig, wenn es mehrere Shop Plugins auf einer Seite gibt und man den Warenkorb mit jeden Klick genau einmal erhöhen möchte.	

Steuerungskonfiguration

Damit können Sie das Verhalten der verschiedenen Ansichtsarten beeinflussen. Fügen Sie das CODE Feld nach dem Wort 'control' ein.

Beispiel:

```
plugin.tt_products.control.LIST.param {
    ignore = pp
}
```

Property:	Data type:	Description:	Default:

param	*array of string*	Hier kann die Bedeutung der Steuerparameter beeinflusst werden. Das wird z.B. benötigt, wenn Sie mehrere Produktlisten auf einer Seite haben, und wenn Sie die URL Parameter deaktivieren müssen, damit in einer Liste nicht ungewollt Produkte ausgefiltert werden, oder an der falschen Position gestartet wird.
		use … wenn Parameter aktiviert ist (nur für backPID)
		item … wenn ein Link Parameter immer aus dem Datensatz des Produktes erzeugt werden muss (für cat)
		
		Beispiel:
		`param {`
		`  ignore = pp`
		`  use = backPID`
		`  item = cat`
		`}`	

Konfiguration für Bezahlung, Versand und Verpackung

Die Bezahlung und der Versand sind einander von der Konfiguration her sehr ähnlich. Daher haben sie im wesentlichen dieselbe Liste an Eigenschaften mit Anmerkungen, wenn eine Eigenschaft nur bei einem zutrifft. Der Konfiguration der Bezahlung und des Versands bietet die Möglichkeit, verschiedene Versandarten und Bezahlmethoden für den Benutzer anzugeben. Außerdem können Sie festlegen, ob es über Auswahlknöpfe oder eine Auswahlbox wählbar sein soll.

Die Nummer der ausgewählten Bezahl- und Versandmethode wird am Ende der Template Datei in der Dankeschön-Seite wieder ausgewertet, damit nur der entsprechende Text dort angezeigt wird . Sie können mit diesen Einstellungen auch die Ausführung von besonderen PHP Skripten veranlassen. Das ist alles möglich.

Beispiel:

```
plugin.tt_products.payment {
    radio = 1
    TAXpercentage = 19
    10.title = Vorkasse
    20.title = PayPal
    20.percentOfTotalShipping = 0.04
    30.title = Nachnahme
    30.price.1 = 4
    30.price.6 = 5.8
    30.showLimit = 99
}

plugin.tt_products.shipping {
    radio = 0
    TAXpercentage = 19
    10.title = Deutschland
    10.image.params = align=absmiddle
    10.price.type = count
    10.price.1 = 2.5
    10.price.20 = 4.8
    10.price.50 = 10
    10.price.100 = 25
    10.price.120 = 30
    10.price.300 = 150
    10.percentOfGoodstotal = 0

    20.title = Europa (nicht Deutschland)
    20.image.params = align=absmiddle
    20.price.type = count
    20.price.1 = 8.8
    20.price.20 = 11.8
    20.price.50 = 11.8
    20.price.100 = 25
    20.price.120 = 30
    20.price.300 = 150
    20.percentOfGoodstotal = 0

    30.title = Selbstabholung
    30.type = pick_store
    30.price.1 = 0
}

plugin.tt_products.handling {
    10 {
      TAXpercentage = 19
      10.title = Druckkosten
      10.price = 17
    }
    20 {
      TAXpercentage = 19
      10.title = Mindestmengenzuschlag
      10.price = 5
    }
    30 {
      TAXpercentage = 19
      10.title = Verpackungskosten
      10.price = 1.95
    }
}
```

Eigenschaft:	Datentyp:	Beschreibung:	Voreinstellung:
radio	boolean	Wenn gesetzt, werden Auswahlknöpfe angezeigt, ansonsten eine Auswahlbox.	0

Eigenschaft:	Datentyp:	Beschreibung:	Voreinstellung:
template	string	**(Nur Layout Auswahlknöpfe)** Wenn .radio gesetzt ist, dann wird dieser Text als 'template' für die Auswahlknöpfe verwendet. **Standard ist (in einer Zeile):** `<nobr>###IMAGE### <input type="radio" name="recs[tt_products]['.$pskey.']" onClick="submit()" value="###VALUE###"###CHECKED###> ###TITLE###</nobr> `	
wrap	string	**(Nur Auswahlbox Layout)** Wenn .radio 0 ist, dann wird dieser Text um die <option> Tags in einem <select>-Tag ausgegeben. **Standard ist (in einer Zeile):** `'<select name="recs[tt_products]['.$key.']" onChange="submit()">|</select>'`	

TAXpercentage	double	Steuersatz.
		Double Wert (!) (das bedeutet, daß der Punkt '.' für die Dezimaltrennung verwendet werden muß)
		Das ersetzt die frühere Eigenschaft priceNoTax. Dieser Steuersatz kann sich von dem globalen mit dem gleichen Namen unterscheiden.
		
		Beispiel:
		`# Dänischer Steuersatz beträgt 25%:`
		`payment.TAXpercentage = 25.00`
			global übernommen

TAXincluded	boolean	Setzen Sie das, wenn die Steuer in den Preisen für Bezahlung/Versand miteingerechnet ist.	global übernommen

Feld von Ganzzahlen		Konfiguration, siehe unten
		
		Beispiels:
		`TAXpercentage = 12`
		`10.title = Credit card`
		`10.image.file = typo3/sysext/cms/tslib/media/logos/dankort.gif`
		`10.price =`
		`10.percentOfGoodstotal = 0`
		`10.calculationScript = EXT:tt_products/pi1/products_comp_calcScript.inc`
		`30.title = By mail`
		`30.image.file = typo3/sysext/cms/tslib/media/logos/postdanmark.gif`
		`30.price = 40`	

Konfiguration der auswählbaren Einträge zu Bezahlung / Versand / Verpackung			
title	string	Name des Eintrags, z.B. "Master Card" oder "Postversand" Der Name wird mehrmals hintereinander aufgelistet, wenn Marker ###STATIC_COUNTRIES_CN_ISO_3### vorkommen und where.static_countries gesetzt worden ist.	
type	string	Nur für Shipping! Art des Versands pick_store … die Produkte im Geschäft abholen. Es werden die Zustelldaten des ausgewählten Ladens eingetragen (siehe UIDstore). nocopy … Die Rechnungsadresse wird nicht in die Lieferadresse kopiert, auch dann nicht, wenn keine Lieferadresse eingetragen worden ist.	
image	IMAGE cObject	Logo Bild für den Eintrag	

price	double oder *Feld von Ganzzahlen siehe unten für weitere Parameter*	Preis eines Eintrags, inkl. oder exkl. Steuer je nach TAXincluded.
		Verwenden Sie Ganzzahlen, um die minimale Anzahl an Produkten festzulegen, für die der Preis gültig ist. 6 Produkte und mehr würden in diesem Beispiel 5.8 Euro kosten.
		Mit dem Typ weight können Sie auch die Versandkosten nach Gewicht der Produkte berechnen.
		
		Beispiel:
		`30.price.type = count`
		`30.price.1 = 4`
		`30.price.6 = 5.8`	

Eigenschaft:	Datentyp:	Beschreibung:	Voreinstellung:
replaceTAXpercentage	double	siehe TAXpercentage Wenn gesetzt, wird die allgemeine TAXpercentage damit überschrieben.	
priceFactWeight	double	**nur Versand:** Der Preis wird aus dem Gewicht aller Produkte berechnet. Das wird zum Gesamtpreis dazugezählt. Das Gewicht wird mit diesem Faktor multipliziert, um den Versandpreis zu erhöhen.	
priceFactCount	double	**nur Versand:** Der Preis wird aus der Anzahl aller Produkte berechnet. Das wird zum Gesamtpreis dazugezählt. Die Anzahl der Produkte im Warenkorb wird mit diesem Faktor multipliziert, um den Versandpreis zu erhöhen.	
percentOfGoodstotal	double	Preis eines Produkts, berechnet über einen Prozentsatz des Gesamtbetrags vor der Bezahlung/Versand	
percentOfTotalShipping	double	**nur Versand:** Wenn gesetzt, werden die Kosten für das Bezahlsystem als Prozente vom gesamten Produktpreis inklusive der Versandkosten und der Steuer errechnet.	
creditcards	string	**nur Bezahlung:** Durch Beistriche/Komma ',' getrennte Liste der uids der erlaubten Kreditkarten. Siehe Datei lgocalland_db.xml (sys_products_cards.cc_type.I) 0 ... American Express 1 ... Diners Club 2 ... Mastercard 3 ... Visa	

Eigenschaft:	Datentyp:	Beschreibung:	Voreinstellung:

creditcardSelect	*Feld von Form Elementen*	**nur Bezahlung:**
		Falls gesetzt, dann werden Monat und Jahr der Kreditkarte als Selectboxen dargestellt.
		Unterwerte: mm (Monat) und yy (Jahr)
		
		Beispiel:
		`creditcardSelect {`
		`  mm {`
		`  label = Month`
		`  valueArray {`
		`  10.label = 1`
		`  10.value = 1`
		`  20.label = 2`
		`  20.value = 2`
		`  30.label = 3`
		`  30.value = 3`
		`  40.label = 4`
		`  40.value = 4`
		`  50.label = 5`
		`  50.value = 5`
		`  60.label = 6`
		`  60.value = 6`
		`  70.label = 7`
		`  70.value = 7`
		`  80.label = 8`
		`  80.value = 8`
		`  90.label = 9`
		`  90.value = 9`
		`  100.label = 10`
		`  100.value = 10`
		`  110.label = 11`
		`  110.value = 11`
		`  120.label = 12`
		`  120.value = 12`
		`  }`
		`  }`
		
		`  yy {`
		`  label = Year`
		`  type = recs[creditcard][endtime_yy]=select`
		`  valueArray {`
		`  10.label = 2007`
		`  10.value = 2007`
		`  20.label = 2008`
		`  20.value = 2008`
		`  30.label = 2009`
		`  30.value = 2009`
		`  40.label = 2010`
		`  40.value = 2010`
		`  }`
		`}`	

accounts	boolean	**nur Bezahlung:** Wenn gesetzt, dann wird die Bezahlung durch Abbuchung vom eingegebenen Bankkonto ermöglicht.	
useAsterisk	Int+	**nur Bezahlung:** Wenn gesetzt, werden die Konto und Kreditkartendaten nach der Eingabe ausgesternt angezeigt.	1
bulkilyAddition	int+	**nur Versand:** Betrag um den die Versandkosten für ein sperriges Produkt erhöht werden.	
bulkilyFeeTax	int+	**nur Versand:** Steuerbetrag in Prozent für das Versenden eines Sperrguts	
calculationScript	resouce	PHP Skript, das inkludiert wird. Es soll die internen Arrays zur Berechnung heranziehen. Dieses Skript kann dazu verwendet werden, um eine besondere Gebühr für Bezahlung/Versand zu berechnen. Für eine Beispielsanwendung siehe pi/products_comp_calcScript.inc, das zeigt wie der Endbetrag um 5.75% seines eigenen Wertes erhöht wird, um die Gebühren der internationalen Kreditkarten Organisation abzudecken. Das Berechnungsskript erhält die TypoScript Eigenschaften im $conf Array.	

Eigenschaft:	Datentyp:	Beschreibung:	Voreinstellung:
handleScript	resource	PHP Skript, das nach dem Abschluß einer Bestellung aufgerufen wird. Dieses Skript muß sich selber um die Bildschirmausgabe und um den Abschluß der Bestellung kümmern. Siehe pi/payment_DIBS.php als Beispiel. Die Eigenschaften des Handle Skripts werden an die Funktion als $conf Array übergeben. Der Inhalt der Variable $content muß den HTML-Inhalt als Ergebnis zurückliefern.	

handleLib	string / array	**nur Bezahlung:**
		Name der TYPO3 Library für das Bezahlungssystem. Im Moment kann hier nur 'paymentlib' gesetzt werden, um die Payment Library Extension von Rober Lemke zu verwenden.
		
		Beispiel:
		`30.handleLib = paymentlib`
		
		Array Werte:
		extName ... Name der spezifischen Payment Library (SPL) Extension
		paymentMethod ... Methode der SPL (z.B. paymentlib_transcentral_cc_mastercard)
		Currency ... zu verwendende Währung
		templateFile ... Template Datei zur Anzeige des Bezahlvorgangs
		gatewaymode ... Modus des Gateways (form / request)
		
		Beispiel:
		`40.title = Mastercard`
		`40.handleLib = paymentlib`
		`40.image.file = EXT:tt_products/res/icons/fe/mastercard.gif`
		`40.handleLib {`
		`extName = transcentral`
		`paymentMethod = paymentlib_transcentral_cc_mastercard`
		`Currency = $`
		`templateFile = EXT:tt_products/template/paymentlib.tmpl`
		`gatewaymode = form`
		`}`	

handleURL	string	Wenn gesetzt, wird dieses handleURL anstelle der Dankeschön-Seite (über PIDthanks) ausgegeben, damit z.B. ein handleScript die Information bzgl. Bezahlsystem usw. Verarbeiten kann.	
handleTarget	string	Alternatives Target der HTML-Form.	

excludePayment	*list of integers*	**nur Versand:**
		Dies ist eine Liste an Schlüsseln der Bezahlmethoden (ihre Nummern), die für eine bestimmte Versandart nicht angezeigt werden. Wenn zum Beispiel Leute Waren im Geschäft kaufen, dann möchten Sie nicht, daß diese Geld überweisen oder Online bezahlen, sondern daß sie die Waren sofort bar bezahlen. Daher können Sie diese Bezahlmethoden ausschließen.
		
		Beispiel:
		`...`
		`  40.title = Abholen der Waren im Geschäft`
		`  40.excludePayment = 10,40`
		`}`	

excludeHandling	*list of integers*	**nur Versand:**
		Dies ist eine Liste an Schlüsseln der Bearbeitungsmethoden (ihre Nummern), die für eine bestimmte Versandart nicht angezeigt werden. Wenn zum Beispiel Leute Waren im Geschäft kaufen, dann möchten Sie nicht, dann benötigen Sie keine transportfähige Verpackung. Daher können Sie diese Bearbeitungsmethode ausschließen.
		
		Beispiel:
		`...`
		`  40.title = Abholen der Waren im Geschäft`
		`  40.excludeHandling = 10,40`
		`}`	

Eigenschaft:	Datentyp:	Beschreibung:	Voreinstellung:

replacePayment	*list of integers*	**nur Versand:**
		Diese Einstellungen der Bezahlung werden überschrieben, wenn diese Bezahlmethode ausgewählt worden ist.
		
		Example:
		`...`
		`  40.title = China`
		`  40.replacePayment.10.title = Bezahlung nach China`
		`  40.replacePayment.10.price = 100`
		`}`	

show	boolean	Wenn das gesetzt ist, wird der Eintrag in der Liste angezeigt.	1
showLimit	double	Wenn gesetzt, dann wird der Eintrag nur dann angezeigt, wenn zumindest diese Anzahl an Produkten im Warenkorb liegt. 0 ... zeige diesen Eintrag immer an	0
type	string	**nur Bezahlung:** fe_users ... die Bezahlart kann über die fe_users Tabelle konfiguriert werden	
visibleForGroupID	int+	**nur Bezahlung:** Diese Bezahlmethode ist nur verfügbar, wenn ein Benutzer eingeloggt ist und wenn er ein Mitglied dieser Benutzergruppe ist.	
addRequiredInfoFields	string	**nur Bezahlung:** Zusätzlich benötigte Felder auf der INFO Seite, wenn diese Bezahlmethode ausgewählt worden ist. Brauchbar für Bezahlungen mit der Kreditkarte.	

where.static_countries	string	**nur Versand:**
		Zum Setzen einer SQL WHERE Bedingung für die mit der Select-Box ausgewählten Länder. Diese stammen aus der static_countries Tabelle der static_info_tables Extension.
		
		Beispiel:
		`plugin.tt_products.shipping {`
		`  10.title = Paket Deutschland`
		`  10.where.static_countries = cn_short_local = 'Deutschland'`
		`  10.price = 5.9`
		`  20.title = Paket EU ###STATICCOUNTRIES_CN_SHORT_DE###`
		`  20.where.static_countries = cn_eu_member = 1 AND cn_short_local != 'Deutschland'`
		`  20.price = 8.9`
		`  30.title = außerhalb EU ###STATICCOUNTRIES_CN_SHORT_DE###`
		`  30.where.static_countries = cn_eu_member <> 1`
		`  30.price = 15`
		`}`	

Configuration price Parameters for payment / shipping			
type	string	Bedeutung der Nummer: count ... die Anzahl der Produkte weight ... das berechnete Gewicht in Gramm price ... der Gesamtpreis der Produkte	

WherePIDMinPrice	*list of integers*	**nur Versand:**
		Setzen eines Minimalpreises für den Versand, wenn es ein Produkt im Warenkorb gibt, das aus dem Sysfolder mit der angegebenen PID stammt.
		Wobei 155 die PID ist und 7.5 ist der Minimalpreis, der als Versandkosten berechnet wird, wenn zumindest ein Artikel im Warenkorb mit der PID liegt.
		
		Beispiel:
		`plugin.tt_products.shipping {`
		`  10.title = Paket`
		`  10.price.type = weight`
		`  10.price.WherePIDMinPrice.155 = 7.5`
		`  10.price.1 = 1.5`
		`  10.price.500 = 2.5`
		`  10.price.1000 =3.5`
		`}`	

Eigenschaft:	Datentyp:	Beschreibung:	Voreinstellung:

calc	array	**nur Versand:**
		zu verwendende Preisberechnungen über Shippingcalc. (siehe Kapitel Versandpreis Berechnung)
		`plugin.tt_products.shipping {`
		`10.title = Paket`
		`10.price.calc {`
		`use = 10`
		`}`
		`}`	

~~productsNoTax~~	~~int+~~	~~**nur Versand:**~~ ~~Wenn gesetzt wird für kein Produkt eine MWSt verrechnet.~~	
mode	string	Modus wie verschiedene Berechnungen summiert werden sollen. mögliche Werte: additive ... Die einzelnen Versandkosten werden aufaddiert.	additive

noCostsAmount	double	Wenn der Gesamtpreis für ein Produkt diese Summe erreicht hat, dann werden keine Kosten berechnet.
		
		Beispiel:
		`plugin.tt_products.shipping {`
		`10.title = Paket`
		`10.price.type = count`
		`10.price.noCostsAmount = 200`
		`10.price.1 = 1.5`
		`}`	

order	string	Durch Komma getrennte Auflistung der Einstellungen zur Preisberechnung. Nach dieser Reihenfolge wird der Preis ermittelt.	

Versandpreis Berechnung shippingcalc/handlingcalc

Die Berechnung des Versandpreises kann noch komplexer durchgeführt werden. Es kann z.B. eine Reihenfolge angegeben werden, nach der die Berechnung durchgeführt wird. Dies ist notwendig, wenn 2 verschiedene Berechnungsarten gleichzeitig verwendet werden. Standardmäßig wird der günstigere Preis verrechnet.

Beispiel:
```
plugin.tt_products.shippingcalc {
   10.type = price
   10.sql.where = category = 12
   10.prod.250 = 0.06
}
```

Property:	Data type:	Description:	Default:

type	string	Eine spezielle Preisberechnung, die nur für bestimmte Produkte gilt. Ident zu discountprice.
		
		percent ... in Prozent des Preises
		
		Im Beispiel wird von allen roten Produkten ein Versandpreis von 25% zum Produktpreis eines roten Produktes aufgeschlagen, immer wenn der Gesamtpreis aller Produkte > 0 ist.
		Beispiel:
		`shippingcalc.10 {`
		`type = price`
		`sql.where = color = 'rot'`
		`prod.type = percent`
		`prod.0 = 0.06`
		`}`	

sql	select	SQL Bedingung über die Datensätze	

Pricecalc, discountprice und creditpoints Konfiguration

Die Preisberechnung (pricecalc) gibt Ihnen die Möglichkeit, Preissummen über die Produkte über eine Berechnungstabelle zu ermitteln. Der Discount Preis (discountprice) wird für Kunden verwendet, die Frontend Benutzer sind und zur Gruppe mit dem Namen gehören, der über discountGroupName gesetzt worden ist..

Eigenschaft:	Datentyp:	Beschreibung:	Voreinstellung:
prod	*Zweischneidige Liste von Ganzzahlen*	Die linke Seite der Integer kennzeichnet Zeilen, die zusammengehören. Die Bedeutung der rechten Integer ergibt sich aus den Einstellungen jeder Zeile. Bei pricecalc gibt es nur für die angegebenen Mengen eine Ermäßigung, bei discountprice auch für alle dazwischen liegenden Stückzahlen ab einer bestimmten Menge.	0

		Pricecalc:
		Spezielle Preise für die Produkte. Wo 1 Produkt 4.99 kostet, dort werden 2 Produkte 8.99 kosten. Mit discountprice wird der Preis für ein Produkt eingegeben, mit pricecalc ist es der Preis für alle Produkte zusammen, für die eines davon 4.99 als Preis im Sysfolder eines x-beliebigen Produktes eingetragen ist. Der discountprice überschreibt die Ergebnisse von pricecalc, weil hier immer das niedrigere Ergebnis als Endpreis herauskommen muß. Wenn das Preisfeld price2 verwendet wird, dann überschreibt dieses alle Preisberechnungen.
		Achtung: getDiscountPrice muss 1 sein, wenn die Rabatte für alle Benutzer gelten sollen.
		
		Beispiel:
		`pricecalc {`
		`  10.type = count`
		`  10.field = price`
		`  10.sql.where =`
		`  10.prod.1 = 4.99`
		`  10.prod.2 = 8.99`
		`  10.prod.5 = 19.99`
		`  20.type = count`
		`  20.field = price`
		`  20.sql.where =`
		`  20.prod.1 = 6.99`
		`  20.prod.2 = 13.98`
		`  20.prod.5 = 29.99`
		`}`	

		Discountprice:
		Hier werden die Preise der Produkte entsprechend der Typ-Einstellung berechnet. Bei 'count' werden die Preise der Artikel zusammengezählt. Die 'additive'-Einstellung bewirkt, daß die Artikel über alle Konfigurations-Zeilen hinweg zusammengezählt werden und nicht wie sonst, nur zeilenweise.
		Verwenden Sie type = count und prod.type=percent, wenn Sie statt Preisangaben lieber Prozentangaben machen wollen. Aber Sie müssen in diesem Fall eine SQL Bedingung setzen, damit nicht alle Produkte dafür herangezogen werden.
		Beispiel:
		`discountprice.50 {`
		`  type = count`
		`  sql.where = color = 'rot'`
		`  prod.type = percent`
		`  prod.0 = 6`
		`}`
		
		Beispiel:
		`discountprice {`
		`  10.type = count`
		`  10.field = price`
		`  10.additive = 0`
		`  10.sql.where =`
		`  10.prod.1 = 4.99`
		`  10.prod.100 = 2.49`
		`  10.prod.1050 = 2.39`
		`  20.type = count`
		`  20.field = price`
		`  20.sql.where =`
		`  20.prod.1 = 6.99`
		`  20.prod.100 = 2.59`
		`  20.prod.1050 = 2.49`
		`}`	

		Creditpoints: Damit wird eingestellt, wieviele Gutscheinpunkte jemand erhalten wird, wenn er Artikel im Shop kauft. Die Werte auf der rechten Seite sind die Prozente der bestellten Artikel, wenn 'type=price' ist. **Beispiel:** `creditpoints {`	

Eigenschaft:	Datentyp:	Beschreibung:	Voreinstellung:
additive	double	Nur für discountprice gültig. Wenn gesetzt, werden alle Produkte von allen Discount-Preisen zusammengezählt, um den effektiven Disount-Preis zu berechnen. Wenn nicht gesezt, dann werden nur die Produkte mit demselben Preis zusammengezählt.	
type	string	Die Bedeutung der Ganzzahl auf der rechten Seite, die normalerweise zusammengezählt wird: count ... die Anzahl der Produkte (nur pricecalc und discountprice) price ... der Gesamtpreis der Produkte (nur creditpoints)	
sql	select	SQL Bedingung über die Datensätze	

pricefactor	double	Damit wird berechnet, wieviel Geld jemand für seine Gutscheinpunkte erhalten wird. 2 Gutscheinpunkte werden 1 Euro an Gutschrift in der eingestellten Währung ergeben.
		
		Beispiel:
		`creditpoints {`
		`  pricefactor = 0.5`
		`}`	

priceprod	double	Preis eines Gutscheinpunktes, der beim Produkt abgespeichert wird. In diesem Beispiel hat ein Gutscheinpunkt den Gegenwert von einem Cent.
		
		Example:
		`creditpoints {`
		`  priceprod = 0.01`
		`}`	

[tsref:(script).productsLib.payment/(script).productsLib.shipping]

Freundschaftswerbung/Gutschein Konfiguration

Die Accounts der Freunde, von denen man angeworben worden ist bzw. die Gutscheinpunkte müssen in einer Datenbanktabelle gespeichert werden. Normalerweise werden in der fe_users Tabelle die Benutzernamen der Frontend Benutzer, von denen jemand auf den Shop aufmerksam gemacht worden ist, abgespeichert. Der werbende Kunde erhält beim nächsten Kauf einen Preisnachlaß. Wenn aber ein Tabellenname eingetragen worden ist, dann läuft es z.B. über die voucher Extension. Es wird dann bei der Eingabe eines gültigen Gutschein-Codes eine Preisermäßigung bewirkt.

Example:

```
plugin.tt_products.voucher {
   table = my_voucher_table
}
```

Property:	Data type:	Description:	Default:
table	string	Name der Tabelle, in welcher die Gutscheinpunkte gespeichert werden. Sie muß ein Feld fe_users_uid mit der uid des FE Benutzers haben. Wenn sie leer ist, wird der Gutschein Code in der fe_users Tabelle im Feld tt_products_vouchercode gespeichert.	
amount	double	Preisermäßigung bei der Verwendung von Benutzernamen als Gutscheine, wenn die fe_users Tabelle verwendet worden ist.	
price	double	Preis für einen Gutschein, wenn er als Belohnung für die Anwerbung eines Freundes verwendet wird.	

Bestellbestätigung Konfiguration

Es gibt einige Einstellungen, die den Ablauf der Bestellbestätigung beeinflussen können.

Example:

```
plugin.tt_products.finalize {
   productsFilter {
      10 {
         pid = 12
         email = myemail@mytypo3site1.com
      }
      20 {
         pid = 12
         email = myemail@mytypo3site2.com
      }
   }
}
```

Property:	Data type:	Description:	Default:
productsFilter	string	Damit werden die Bestellbestätigungsemails auf die Produkte unterschiedlicher Seiten aufgeteilt. Die jeweiligen Emails erhalten nur die Produkte aus den angegebenen Seiten. Das Array enthält diese Einträge: pid ... eine Liste von Email email ... eine Liste von Email Adressen	

Rechnung und Lieferschein Konfiguration

Es gibt einige Einstellungen, die bei der Erstellung und der Verwendung von Rechnung und Lieferschein nützlich sein können. Sie werden im Verzeichnis "fileadmin/data/bill" erstellt.

Example:

```
plugin.tt_products.bill {
   generation = auto
}
plugin.tt_products.delivery {
   generation =
}
```

Property:	Data type:	Description:	Default:
generation	string	Damit wird die Erzeugen von Rechnung oder Lieferschein festgelegt. auto ... Die Erzeugung wird automatisch beim Abschluss der Bestellung durchgeführt.	
type	string	Typ von Rechnung oder Lieferschein html ... HTML Format pdf ... PDF Format (Die Extension 'fpdf' muss installiert sein.)	

Bestellung/Verwaltung Konfiguration

Mit diesen Einstellungen kann die Anzeige des Tracking geändert werden.

Beispiel:

```
plugin.tt_products.tracking {
   fields = uid,name,tracking_code,amount,status,status_log,bill_no
}
```

Eigenschaft:	Datentyp	Beschreibung:	Voreinstellung:
fields	string	Damit werden die Felder der Tabelle sys_products_orders festgelegt, die in der Auswahlbox der Bestellungen angezeigt werden.	

recordLine	string	Vollständige HTML Zeile für den Optionen Tag der Bestellungen in der Select-Box.
		
		Beispiel:
		`plugin.tt_products.tracking {`
		`recordLine = ###ORDER_UID### (###ORDER_BILL_NO###): ###ORDER_NAME### (###ORDER_AMOUNT### ###CUR_SYM###) / ###ORDER_STATUS###) ###ORDER_CRDATE### ###LAST_ORDER_TYPE### ###LAST_ORDER_COUNT###`
		`}`	

recordType	string	Typ der HTML Auswahlbox für die Bestellungen: • select • radio • checkbox	

Eigenschaft:	Datentyp	Beschreibung:	Voreinstellung:

recordBox	->stdWrap	Box, welche die Bestellzeilen umgibt.
		Beispiel:
		`plugin.tt_products.tracking {`
		`recordBox.wrap = <table id="thetable" cellspacing="0"><tbody>`
		`| </tbody></table>`
		`}`	

sql.where	string	SQL Where Bedingung für die Auswahlbox der Bestellungen. Die Bedingung für die versteckten und gelöschten Dateien wird automatisch angehängt.
		Beispiel:
		`plugin.tt_products.tracking {`
		`sql.where = status!=0 AND status<100`
		`}`	

sql.orderBy	string	SQL Sortierung für die Auswahlbox der Bestellungen.
		Beispiel:
		`plugin.tt_products.tracking {`
		`sql.orderBy = crdate`
		`}`	

view

payment_DIBS.php

Eigenschaften des handleScript für eine Schnittstelle zu DIBS (http://www.architrade.com/uk/):

Eigenschaft:	Datentyp:	Beschreibung:	Voreinstellung:
templateFile	resource	Template Datei für die Verwendung mit DIBS Sie müssen folgende Zeile in die FORM des tt_products Templates eintragen, bevor das DIBS Skript aufgerufen wird: `<input type="hidden" name="products_cmd" value="cardno">`	
soloe	boolean	Wenn gesetzt, verwendet dieses Skript Unter-Templates mit der Markierung ###DIBS_SOLOE_TEMPLATE### anstelle der standardmäßigen Markierung ###DIBS_CARDNO_TEMPLATE###	
direct	boolean	Wenn gesetzt, verwendet dieses Skript Unter-Templates mit der Markierung ###DIBS_DIRECT_TEMPLATE### anstelle der standardmäßigen Markierung ###DIBS_CARDNO_TEMPLATE###	
merchant	boolean	Händler id	
currency	int+	Währungsnummer, ISO4217 Format	
relayURL	string	Die Url des Shop dessen gesicherter Server den Warenkorb übernimmt.	
test	boolean	Wenn gesetzt, wird das Test-Feld in der FORM gesetzt.	

cardType	string	Kartentyp,
		Beispielwerte:
		`DK = Dankort`
		`V-DK = Visa-Dankort`
		`MC(DK) = Mastercard/Eurocard issued in Danmark`
		`VISA = Visakort issued abroad`
		`MC = Mastercard/Eurocard issued abroad`
		`DIN(DK) = Diners Club, Denmark`
		`DIN = Diners Club, international`	

account	string	DIBS Kontoeigenschaft	
addOrderInfo	boolean	Wenn gesetzt, werden die Bestelldaten an die FORM angehängt. DIBS kann diese Information auswerten und zusammen mit den Bezahlinformationen anzeigen.	
k1 k2	string	DIBS Schlüsselwerte	

[tsref:(script).productsLib.paymentDIBS]

Bekannte Probleme

Prüfliste:

- füge unter 'Template' – Static (aus Extensions) die statische Template Datei 'Shop System' ein.
- **niemals** das 'plugin.tt_producs [DEPRECATED]' im Setup verwenden
- setze 'auf Lager' für alle Produkte und Artikel auf einen Wert größer 0
- verwende keine TYPO3-Einstellungen zur Mehrsprachigkeit oder lege Einträge in den Tabellen 'Produkte – alternative Sprache' an.
- die Shop Template Datei muss gefunden werden
- ein Fehler im Shop erzeugt einen Eintrag in der PHP error_log Datei (im Install Tool und php.ini aktivierbar)
- der Cache ist vorher gelöscht worden
- setze pid_list und recursive im TypoScript Setup. Setze andernfalls den Ausgangspunkt und Rekursiv im Plugin, damit alle Seiten des Shops erfaßt werden.

Allgemein:

– Wenn Sie über den Extension Manager ein Update von einer vorher installierten älteren Version von tt_products machen, und wenn Sie vorher nicht die Table Library 'table' und FEBE Library 'fh_library' installiert haben, dann wird das ganze TYPO3 CMS unbenutzbar und Sie erhalten die folgende Fehlermeldung

TYPO3 Fatal Error: Extension key "table" was NOT loaded! (t3lib_extMgm::extPath)

Schritte zur Reparatur des TYPO3 CMS:

```
1. Editieren Sie die Datei typo3conf/localconf.php
2. Löschen Sie den Eintrag ',tt_products'
3. Löschen Sie die temp_CACHED_ Dateien aus typo3conf
4. Klicken Sie auf Neuladen im Internet Browser
```

– Verwenden Sie immer die Versionen der Table Library und fh_library, wie in tt_products im Extension Manager angegeben.

– Bitte sehen Sie auf die Webseite http://wiki.typo3.org/index.php/Ext_tt_products .
Oder schauen Sie auf http://jambage.com nach.

– .htaccess muß richtig konfiguriert werden, sonst läuft tt_products nicht wie erwartet, weil das Plugin sonst nicht alle benötigten Skripte finden kann.

– **Falscher Parameter, GET/POST var 'tt_products[product]' wurde nicht angegeben oder kein Produkt mit uid = 0 gefunden.**

Sie müssen die Einstellungen für PIDitemDisplay vornehmen, damit der Shop weiß, wie der Linkparameter 'tt_products[product]' auf die Einzelansicht erzeugt werden muß.

– **Achtung Massen-Abmahnungen!**
Rechtliches: http://www.wettbewerbszentrale.de/media/getlivedoc.aspx?id=567
Fehler im Online-Handel: http://www.wettbewerbszentrale.de/media/getlivedoc.aspx?id=568
Beim Versand ins Ausland müssen Sie auch die Gesetze aller Länder genauestens beachten, in die Sie Waren versenden. Damit ersparen Sie sich sehr hohe Kosten verursacht durch saublöde Abmahnungen eines Anwalts, der auch mehrere Filialen gleichzeitig vertreten könnte, womit sich solche Abmahnungen für einen einzigen Fall sogar multiplizieren könnten. Außerdem könnte für jedes betroffene und falsch ausgezeichnete Produkt eine eigene Abmahnung verschickt werden, also eine weitere Multiplikation der Fälle durch den gegnerischen Anwalt vorgenommen werden. Im Internet bestimmt der Abmahner den Gerichtsort danach, wo er bisher von den Gerichten die höchsten fiktiven Schadenssummen zugesprochen erhalten hat.

– Das Plugin zeigt eine leere Seite
Siehe die Prüfliste oben.

To-Do Liste

- Wunschliste: http://wiki.typo3.org/index.php/Ext_tt_products#Wishlist
- ab 2.7.0: PHP5 ist Voraussetzung

Changelog

siehe auch Datei tt_products/ChangeLog

- 26.03.2005 Einbringen des Codes von Klaus Zierer (zk_products), mehrspaltige Listenansicht, Rechnung, Lieferschein und spezielle Preisberechnungen von Franz Holzinger
- 06.05.2005 zweiter Preis und Preis für Zubehör von Jens Schmietendorf, Beispiel Template von http://bananaguard.de , zk_products 1.3.2 von Klaus Zierer, MWSt von Franz Holzinger
- 23.06.2005 products_mail.inc gelöscht, Beispiel Template von Franz Holzinger
- 26.07.2005 Anzeige der Bestellungen, Gutscheinpunkte System und Freundschaftswerbung von Bert Hiddink
- 03.08.2005 Anmerkungen aus dem zk_products Forum, PIDtracking
- 11.09.2005 Geschenk Gutscheine -14.09.2005 Flexforms statt CODEs
- 05.10.2005 Ergänzungen von Andrea Herzog-Kienast (--> Übersetzer für das französische Manual gesucht)
- 13.10.2005 Gutscheinpunkte und Freundschaftswerbung
- 28.10.2005 Zubehör ist durch Größe mit Eintrag in der Artikeltabelle ersetzt worden.
- 14.07.2006 Ab jetzt muß das statische Shop-Template manuell im Template Setup eingebunden werden. CSS-Shop Template von Robert Markula für ein barrierefreies Shop
- 06.10.2007 Das CODE Feld wird im Seiten Modul für die ausgewählten Flexforms angezeigt.